NASA/CR—2003-212357 CFDRC PR 4116/1

SCISEAL: A Computer Program for Study of Fluid Dynamic Forces in Seals

Users' Manual

M.M. Athavale and A.J. Przekwas
CFD Research Corporation, Huntsville, Alabama

Restriction changed to Unclassified/Unlimited on July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

~~**Export Administration Regulations (EAR) Notice**~~

~~This document contains information within the purview of the Export Administration Regulations (EAR), 15 CFR 730–774, and is export controlled. It may not be transferred to foreign nationals in the U.S. or abroad without specific approval of a knowledgeable NASA export control official, and/or unless an export license/license exception is obtained/available from the Bureau of Industry and Security, United States Department of Commerce. Violations of these regulations are punishable by fine, imprisonment, or both.~~

October 2005

NASA/CR—2003-212357 CFDRC PR 4116/1

SCISEAL: A Computer Program for Study of Fluid Dynamic Forces in Seals

Users' Manual

M.M. Athavale and A.J. Przekwas
CFD Research Corporation, Huntsville, Alabama

Restriction changed to Unclassified/Unlimited on July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

~~**Export Administration Regulations (EAR) Notice**~~

~~This document contains information within the purview of the Export Administration Regulations (EAR), 15 CFR 730–774, and is export controlled. It may not be transferred to foreign nationals in the U.S. or abroad without specific approval of a knowledgeable NASA export control official, and/or unless an export license/license exception is obtained/available from the Bureau of Industry and Security, United States Department of Commerce. Violations of these regulations are punishable by fine, imprisonment, or both.~~

Prepared under Contract NAS3–25644

National Aeronautics and
Space Administration

Glenn Research Center

October 2005

Acknowledgments

Support and contributions of the following individuals in the development of the present code is acknowledged: Mr. R.C. Hendricks (Technical Advisor, NASA GRC) and Dr. A.K. Singhal (Technical Director of CFDRC) for overall guidance of the code development; Dr. Y.G. Lai, Dr. R.K. Avva, and Dr. H.Q. Yang for participation in algorithm and physical model development; and Ms. J.L. Swann for preparation of this report.

Document History

This report was originally published by CFD Research Corporation in May 1995.

Document Availability Change Notice

NASA/CR—2003-212357

SCISEAL: A Computer Program for Study of Fluid Dynamic Forces in Seals, Users' Manual

M.M. Athavale and A.J. Przekwas

This document was published in July 2003 with the restriction Export Administration Regulations (EAR). It was changed to Unclassified/Unlimited July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

Per the STI Program Office and Code I at Headquarters, you may modify copies in your possession. The restriction notice on the cover, title page, and Report Documentation Page should be boldly crossed out and the above statement printed clearly above or below it.

Trade names or manufacturers' names are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

This work was sponsored by the Low Emissions Alternative Power Project of the Vehicle Systems Program at the NASA Glenn Research Center.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at http://gltrs.grc.nasa.gov

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

PREFACE

The Seals Scientific Code (SCISEAL) is an advanced Computational Fluid Dynamics (CFD) package, designed for the calculations of turbulent flows and fluid dynamic forces in different types of seals such as cylindrical, labyrinth, face, and tip seals. The code was developed by CFDRC for NASA Lewis Research Center under a subcontract from Mechanical Technologies, Inc. (MTI). It is part of a five-year (1990-1995) effort planned for the development and validation of a knowledge based system (KBS) for seal design. The KBS consists of:

- Industrial codes; and
- Scientific code.

This report gives complete documentation of the implicit multi-domain SCISEAL code. Several, seal specific capabilities have been built in it for automatic grid generation and rotordynamics of cylindrical seals. The code can also be used to treat fluid flows by setting up appropriate input files using this manual. The initial single-domain SCISEAL was developed from REFLEQS code, developed for NASA MSFC by CFDRC personnel. Several basic algorithm improvements (such as colocated variables, moving grids, rotating coordinates, etc) were made to REFLEQS, and the seal specific features for cylindrical seals were added. The basic code capability was then extended to an implicit multi-block (multi-domain) formulation to facilitate efficient solutions on seals problems with discrete domain blocks such as in labyrinth seals.

This report provides guidance to the users of SCISEAL in problem definition and input file preparation for running it on SCISEAL. The report describes the code structure, model and grid generation, boundary conditions and several seal specific features. Sample input files are also provided in an Appendix which can serve as a good starting point for different types of seals. The SCISEAL code is advanced and general so that it can be used to compute flow in a wide variety of flow problems besides seals. This manual provides a fairly detailed description of the general parameters needed to set up input files and provides some guidelines for the user to avoid commonly made mistakes and provides some tips for obtaining a solution and optimizing code performance. The manual also lists a number seal-specific commands. Thes appear in the appropriate sections with special mention made on how to use these for seals.

Support and contributions of the following individuals in the development of the present code is acknowledged:

- Mr. R.C. Hendricks (Technical Advisor, NASA LeRC) and Dr. A.K. Singhal (Technical Director of CFDRC) for overall guidance of the code development;
- Drs. Y.G. Lai, R.K. Avva, and H.Q. Yang for participation in algorithm and physical model development; and
- Ms. J.L. Swann for preparation of this report.

SECTION 1

INTRODUCTION

1.1 Purpose and Outline of the Report

This report documents the capabilities of the Scientific Seals Code, SCISEAL, and provides guidelines and rules for preparing input files needed by the pre-processor, SCIPRE for eventual execution on SCISEAL. This manual is intended for users with access to a workstation and a text editor. Similar files can also be generated using the KBS (Knowledge-Base System), developed as part of the overall SEALS contract, and which uses IBM PC/workstations running OS/2 operating systems.

The SCISEAL code originally was developed from REFLEQS-3D, a single-domain Navier-Stokes flow solver written by the staff of CFDRC. Subsequently the multi-domain flow solution capability was added to this code bringing it to its present form.

The remainder of this section (Section 1) summarized SCISEAL capabilities and limitations, physical model solution schemes and the seal-specific features built in the code. Section 2 describes the code structure, the pre-processor, SCISEAL, and the steps needed to execute SCISEAL to obtain a solution. Also described are the various output files generated by the code. Section 3 outlines the structure of an input file and the various sections needed to complete problem description. Sections 4 through 12 describe each of the sub-sections in the input file and how to write commands in each section to describe the flow problem. The final section provides some guidelines for use of the code and some common problems to be avoided. This is expected to aid the user in preparation of the input files as well as successful execution of the SCISEAL code.

1.2 Capabilities of the Code

The Seals Scientific code solves the full Navier-Stokes equations for fluid flows in a generalized coordinate system. The code has been equipped with several advanced physical models, numerical schemes and convenient means of problem specifications. The final version of SCISEAL has the following features.

a. Cartesian, axisymmetric, and boundary fitted coordinates
b. an implicit, multi-domain grid solution capability;
c. 3-D, 2-D planar, and 2-D axisymmetric flow treatment;
d. Automated geometry specifications and grid distribution for cylindrical seals;
e. Porosity-resistivity technique for flows with internal objects;
f. Arbitrary placement of boundary conditions;
g. Compressible and incompressible flows;
h. Steady state and transient calculations;
i. High- and Lo- Reylonds number turbulence models, the two-scale turbulence model of C.P. Chen, and a 2-layer model;
j. Fully implicit and conservative formulation;
k. Symmetric whole field solvers and a conjugate gradient solver;
l. Pressure-based solution algorithms, a variation of SIMPLEC and PISO;
m. A choice of high-order differencing schemes, including a 2nd order central-differencing and several upwind -biased 3rd order schemes in space, and upto 2nd order accurate schemes in time; and
n. Moving grid solution methodology for time-accurate solutions on deforming grids.

In addition, there are several features present in the code that apply to cylindrical seals. These include:

a. Simplified single line commands for grid generation with appropriate grid clustering.
b. Automated rotordynamic coefficient calculation procedures. These include: (1) circular whirl orbit method based on full CFD solutions

useful for centered seals; and (2) a small perturbation Navier-Stokes method based on solution of perturbation variables is also available and is the choice for eccentric seals.

c. Post-processing routines to calculate integrated quantities such as rotor load, torque, and frictional power loss, and attitude angle for bearings.

1.3 Code Qualifications and Validation

The SCISEAL code has been checked for the following:

a. Self-consistency of solutions, *i.e.* mass, momentum, and energy conservations;

b. Convergence to unique solutions irrespective of initial (guessed) conditions;

c. Speed and reliability of convergence under different geometric and flow conditions; and

d. Sensitivity of solutions to different geometrical, physical, and flow parameters.

The code has undergone, and is undergoing validation on a continuous basis against several benchmark problems including:

1. Three-Dimensional Flow in a Lid Driven Enclosed Cubic Cavity; data reported by Freitas, *et al.* from Stanford University.
2. Three-Dimensional Flow between Eccentric Rotating Cylinders.
3. Developing Laminar Flow in a 90° Bend Square Duct; experiment of Humphrey, Taylor and Whitelaw, *Journal of Fluid Mechanics*, 1977.
4. Developing Laminar Flow in a 180° Bend Square Duct; experiment of Hille, Vehrenkamp, and Schulz, *Journal of Fluid Mechanics*, 1985.
5. Turbulent Jet-in-Crossflow; experimental data of Crabb, Durao, and Whitelaw, ASME *Journal of Fluid Engineering*, 1981.
6. Forced Convection and Heat Transfer in 90° Curved Diffuser; experimental data of Rojas, Whitelaw, and Yanneskis, ASME Journal of Heat Transfer, 1987.

A comprehensive list of validation/demonstration problems relevant to seals is included in the Interim Report for September 1993. This list is updated as additional problems are simulated using the scientific code.

1.4 Physical Models

The SCISEAL code has provisions for the following turbulence models.

- k-ε turbulence model;
- Multi-scale turbulence model with variable partition of energy spectrum;
- Low Reynolds number turbulence model; and
- 2-layer, k-ε model.

A wall roughness treatment is also available in conjunction with the k-ε turbulence model and the 2-layer k-ε model.

1.5 Numerical Schemes

The SCISEAL code employs the following numerical schemes.

- A pressure-based, finite-volume solution method;
- A colocated arrangement of velocity and scalar variables;
- An implicit, multi-domain formulation that allows the breakup of the flow domain into sub-domains for optimum grid point use and efficient execution;
- A fully implicit formulation, for efficient simulation of transient flows;
- Up to third accurate differencing schemes: Upwind, hybrid, central differencing (with damping), Smart, Osher-Chakravarthy, VanLeer's Minimod and Superbee schemes.
- Iterative, whole-field solutions of dependent variables [each variable is solved for over the whole calculation domain (rather than over a plane, line, or point) before the solution of the next variable starts];
- Use of CFDRC's latest whole-field equation solver with extended,

symmetric links (the use of an efficient and robust linear equation solver is important for the overall computational efficiency of solutions). An efficient conjugate-gradient linear equation solver is also available.

- Use of a modified SIMPLEC algorithm; and
- Use of dynamic storage and vectorizable coding practices.
- Strongly conservative body fitted coordinate formulation.

SECTION 2

CODE STRUCTURE

The Seals Scientific Code consists of two separate computer modules: the pre-processor, SCIPRE, and the flow solver, SCISEAL. Communication between the two modules is carried out by way of intermediate disk files: model.DAT and model.AUG. The generic word "model" in the two files can be replaced by the user with other appropriate words for problem identification. The two files are then read by SCISEAL (Figure 2-1) and contain all information needed by SCISEAL to simulate a fluid flow problem. The main processor, SCISEAL, generates several files, outlined in an Appendix, which include a formatted output file, a binary file for restart data as well as PLOT3D compatible data files for data viewing.

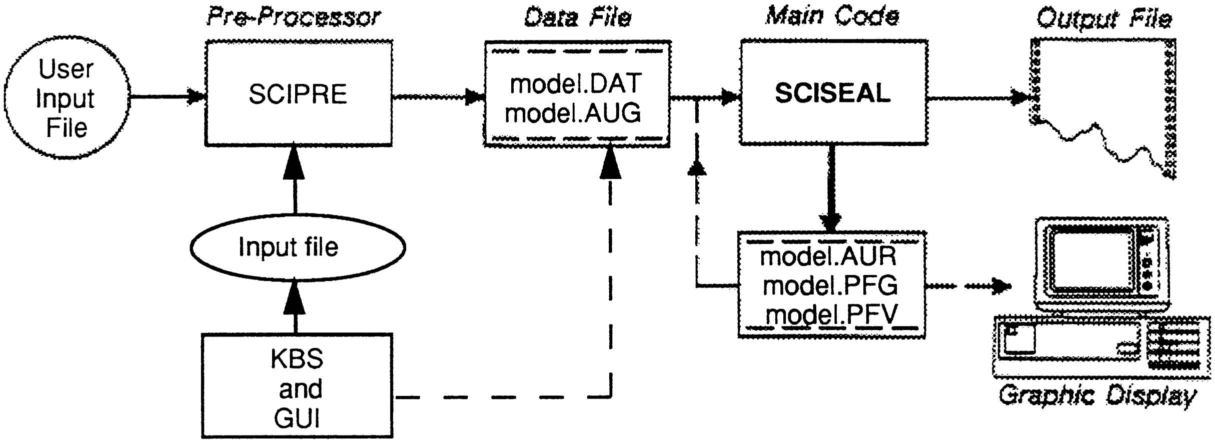


Figure 2-1. Structure of the Seals Code

2.1 The Pre-Processor

The pre-processor, SCIPRE, is an independent module whose main functions are to read the user input data file with a set of input data commands, check them for consistency and correctness, and generate a disk data file (model.DAT) for SCISEAL. The information includes physical properties, initial and boundary conditions, and instructions for the selection of appropriate physical and numerical models to solve the problem. The pre-processor will also generate an unformatted grid file, model.AUG that contains grid type and grid vertex coordinates. A flow chart of the pre-processor is shown in Figure 2-2.

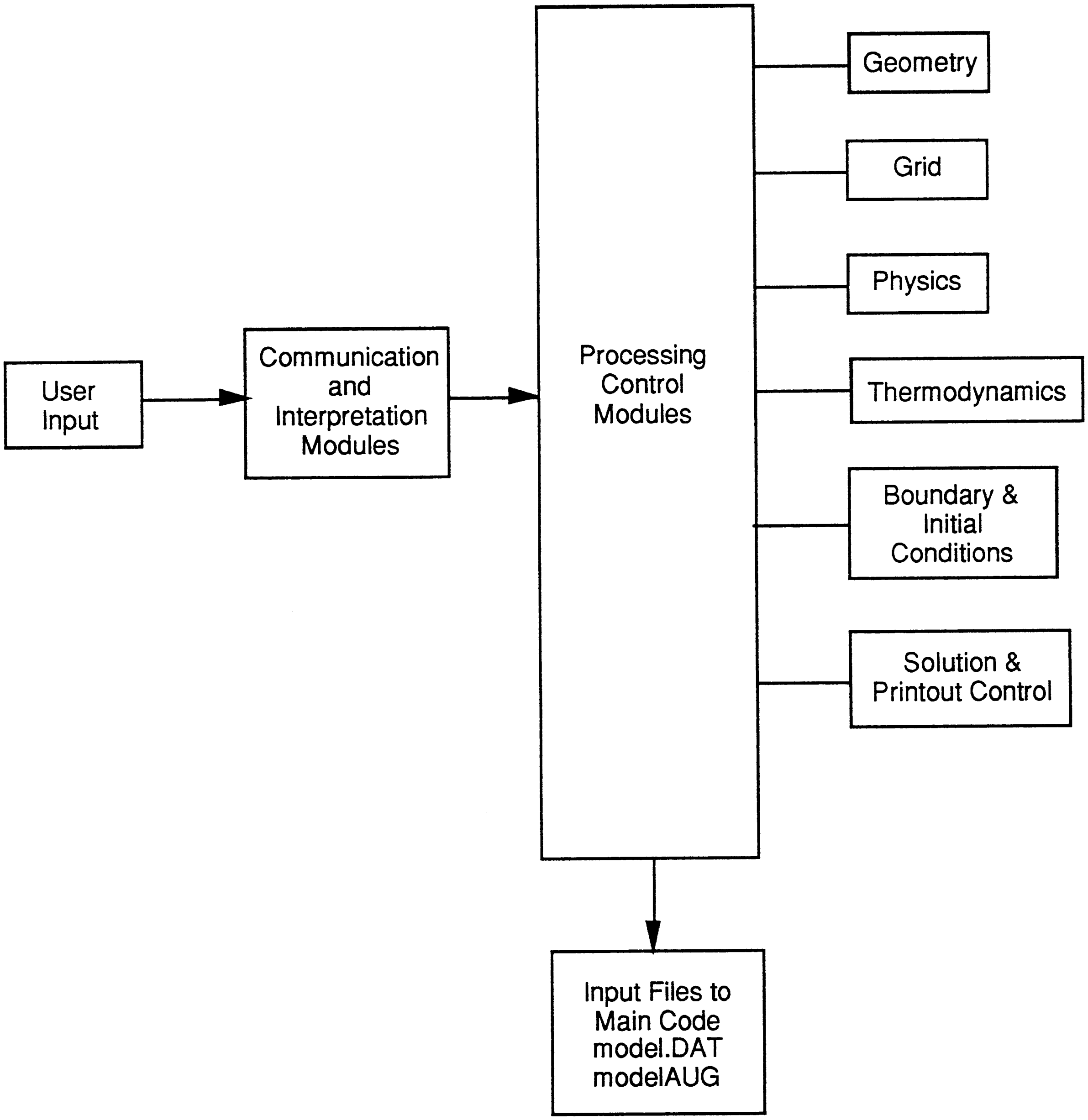

User Input
Communication and Interpretation Modules
Processing Control Modules
Geometry
Grid
Physics
Thermodynamics
Boundary & Initial Conditions
Solution & Printout Control
Input Files to Main Code model.DAT modelAUG

Figure 2-2. Structure of the Pre-Processor, SCIPRE

The pre-processor consists of several major modules, with several subroutines in each module. The modules are arranged by the function each serves. A brief description of the functions of the major modules is given below.

1. **I/O Module**: controls the interaction between the user input file and pre-processor. The input file is read a line at a time and processed for subsequent reading by other modules.

2. **Main Module**: directs the controls to the modules depending on the action requested. Also controls the code size, consistency checks, and writing of disk files.

3. **Boundary Conditions Module**: imposes user specified as well as default boundary conditions; checks for consistency.

4. **Models Module**: assigns various physical models being used in the flow problems.

5. **Solution Control Module**: assigns control parameters needed during the iterative solution procedure, such as solvers, iterations etc.

6. **Grid Generation**: this module allows grid generation using internal algebraic grid generation package, as well as allows processing of externally created grids.

In addition, other modules are used for functions such as defining the local variables, specification of properties and specific seal grid generation.

2.2 Input Data Structure

The pre-processor SCIPRE accepts input data for a problem in the form of a text file. The input file is arranged in the form of several blocks of information delivered bya command at the start of the block and the word END by itself on a line. All of the relevant information for each block is contained between these two lines. There is some flexibility in the order of which these blocks can appear and the structure of the input file is given in the next section (Section 3), together with a sample input file for a 3-D cavity flow problem.

2.3 Structure of the Main Code

The main code (equation solver) includes a large number of modules, each with specific functions such as treatment of boundary conditions, grid and geometry, initial field, physical properties, physical models, solution algorithms and output control. Most of the physical and numerical models are coded in self-contained, modular subroutines, so it is easy to update.

Figure 2-3 illustrates the structure of the main code and indicates the relevant external files used as input or created as a result of the code execution. The files created by the pre-processor, model.DAT and model.AUG, are read by the main code before any computations start. For a restart run, a model.AUR data file, created by the previous computations, is also read. After successful job completion, two new files are created: 1) the output file with the convergence history and user specified field printout, and 2) the restart file that can be used for further graphical post-processing or further restart runs.

Depending on the options requested for seal specific quantities, several small text files are also created by the code. These include:

a. **stload.out** file for steady state rotor load, torque, and power loss;

b. **kctrn.out** rotordynamic coefficient data with circular whirl orbit; and

c. **kcpert.out** rotordynamic coefficient data with small perturbation method.

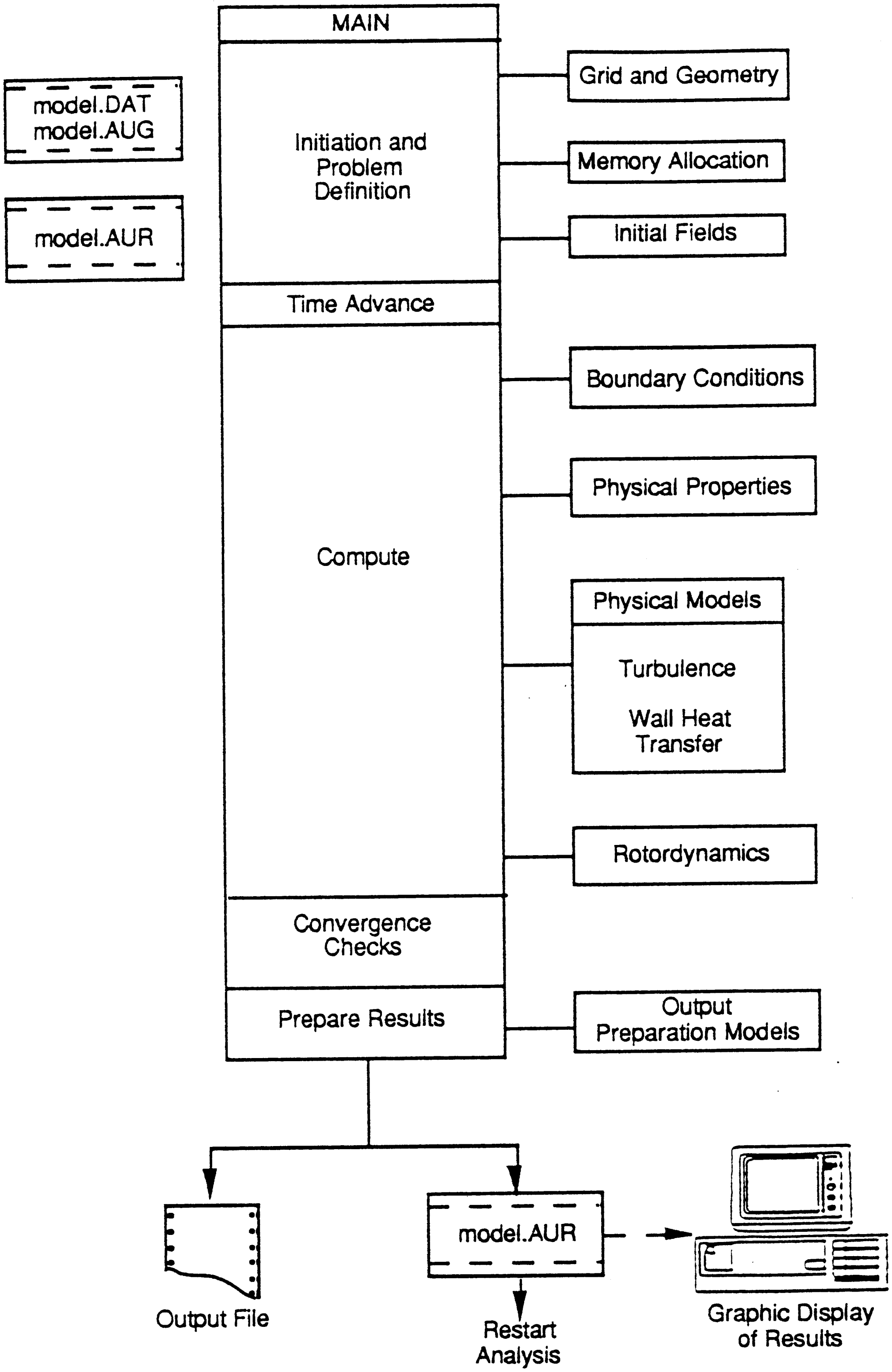


Figure 2-3. Structure of the Main Flow Solver

SECTION 3

INPUT FILE FORMAT

A general description of the SCISEAL command language file is provided in this section. First the overall format of the file is described, followed by the conventions for command language statements and a note about the physical units used in the commands. Finally, a sample input file for a 3-D problem is provided to illustrate the structure of the input file.

3.1 Structure of the Input File

The command language file consists of two optional statements identified by the keywords TITLE and MODEL followed by ten sections which begin with specified keywords and end with the word END on a single input line. The format is illustrated below.

```
TITLE 'Problem Title (up to 72 characters)'
MODEL model-name
PARAMETERS

END
FUNCTIONS

END
GEOMETRY

END
PROBLEM_TYPE

END
PROPERTIES

END
MODELS
```

```
END
BOUNDARY_CONDITIONS

END
INITIAL_CONDITIONS

END
SOLUTION_CONTROL

END
OUTPUT

END
```

The TITLE statement allows input of a problem title that will be printed in the analysis results file. The text string is input following the TITLE keyword and must be enclosed within quotation marks. Care should be taken not to include any quotation makrs in the title line other than the starting and ending quotes, as it would generate an error message. The MODEL statement allows input of a 'model name' which will be used to set the names of the intermediate files created by the Preprocessor and Analysis Modules. If no MODEL statement is provided then the default name 'MODEL' is used.

There are specific commands which are allowed within each section of the input file and these commands are described in the next ten sections.

Conventions

Command language statements are provided in free format using blanks to delimit the various keywords and parameters. The general format of command descriptions in this manual is

COMMAND KEYWORD$_1$ = value$_1$ KEYWORD$_2$ = value$_2$ [KEYWORD = value$_3$]

COMMANDs and KEYWORDs, shown in upper case, are to be entered exactly as presented. The lower-case values are to be replaced with user-supplied input. Elements enclosed in brackets are optional. Multiple KEYWORDS in a vertical list within brackets represent mutually exclusive options. Unless stated otherwise, the first element of the list is the default setting.

The pre-processor does not distinguish between upper-case and lower-case letters so there is no need to use only upper case letters. Multiple commands can be input on a single line of the input file by separating each command with a semicolon. Comment statements, which begin with an asterisk, may be inserted anywhere in the input file.

3.2 Units

SCISEAL employs SI units. All lengths must be given in meters, mass must be specified in kilograms, time must be specified in seconds and temperature must be given in Kelvin. Angular measures are generally input in degrees unless noted otherwise.

The following sub-sections describe a simple 3-D flow problem and the input file that can be used to define the problem and obtain a solution from SCISEAL. The problem selected here is the flow in a "lid-driven cubic cavity" considered in the CFD community as a typical benchmark test case.

3.3 Problem Description

Figure 3-1 shows the configuration of the driven cavity. The geometry consists of a cubic box enclosed by six walls with the top wall moving horizontally with a constant velocity. Due to symmetry of the problem, only half of the problem in the z direction is analyzed.

A uniform 10 × 10 x 5 grid, as shown in Figure 3-2, is used in this demonstration test case.

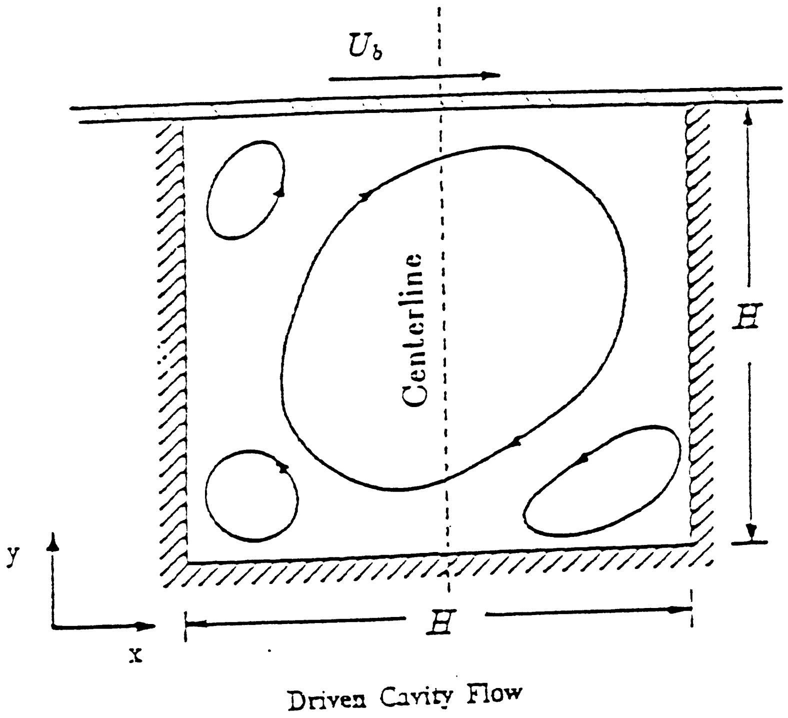


Figure 3-1. Schematic of the Sample Problem

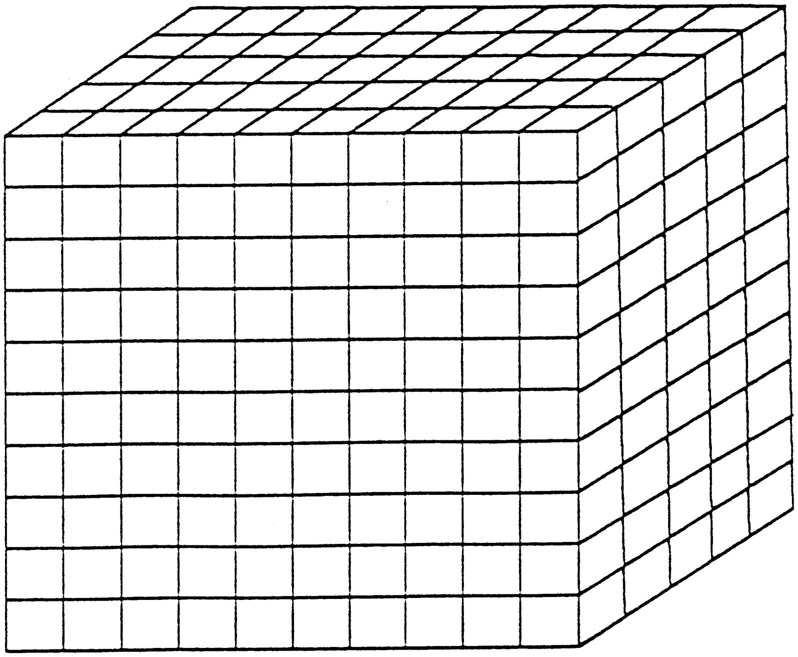


Figure 3-2. Computational Grid for the Sample Problem

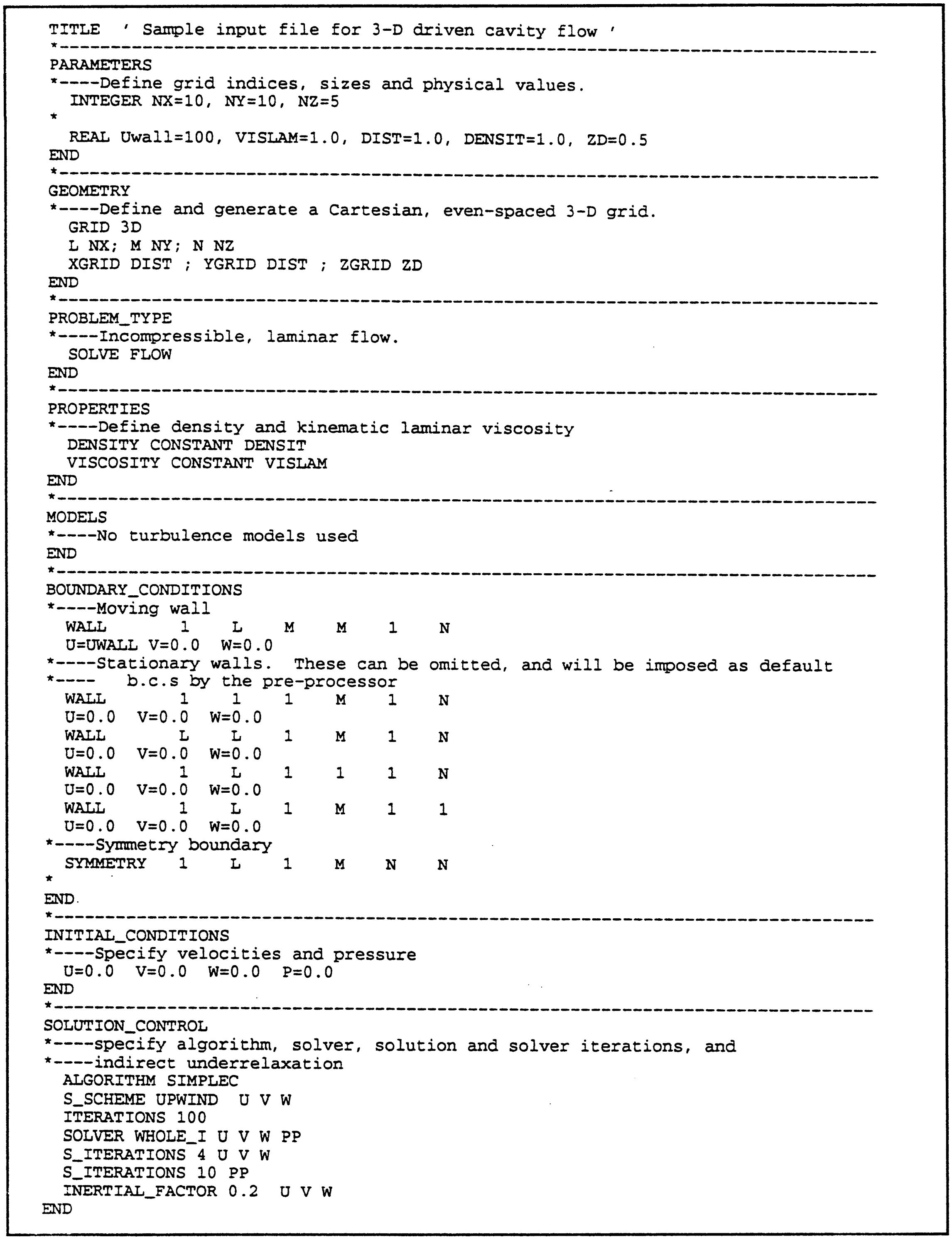
3.4 Input Data Presentation

Figure 3-3 shows the sample input data file for the driven cavity problem. The problem title is specified through the **TITLE** command. The same title also appears in the output file in the problem identification section. Then some integer and real parameters are defined for later use in the input data commands. Five commands are used to set up the grid and geometry. **L**=NX, **M**=NY and **N**=NZ define the number of cells in x-, y, and z-coordinate directions. The keyword 3D in geometry section denotes a 3-dimensional problem with a default Cartesian grid. A uniform grid in the x- and y-directions is specified with the DIST being the square box boundary line length.

The number of iterations is set to 100. Only u-velocity, v-velocity, w-velocity and pressure correction (PP) are solved in the problem. SIMPLEC solution algorithm and upwind differencing scheme have been used for all dependent variables.

A constant density and viscosity is specified as shown in Properties section. The values of density and viscosity are set to DENSIT and VISLAM. DENSIT and VISLAM were previously set to 1. and 0.1, respectively, by the command **REAL**, in the Parameters section.

The domain has six wall boundaries, and are defined using the limiting cell indices next to the boundaries. Based on the indices the pre-processor will impose the boundary on the correct cell faces. The "top" wall is moving with a velocity v_b defined in Parameter list. All other walls are stationary. The pre-processor also has a facility where any unspecified boundaries are taken as stationary walls and this default could have been activated if the stationary walls had not been specified.



```
TITLE  ' Sample input file for 3-D driven cavity flow '
*------------------------------------------------------------------------------
PARAMETERS
*----Define grid indices, sizes and physical values.
  INTEGER NX=10, NY=10, NZ=5
*
  REAL Uwall=100, VISLAM=1.0, DIST=1.0, DENSIT=1.0, ZD=0.5
END
*------------------------------------------------------------------------------
GEOMETRY
*----Define and generate a Cartesian, even-spaced 3-D grid.
  GRID 3D
  L NX; M NY; N NZ
  XGRID DIST ; YGRID DIST ; ZGRID ZD
END
*------------------------------------------------------------------------------
PROBLEM_TYPE
*----Incompressible, laminar flow.
  SOLVE FLOW
END
*------------------------------------------------------------------------------
PROPERTIES
*----Define density and kinematic laminar viscosity
  DENSITY CONSTANT DENSIT
  VISCOSITY CONSTANT VISLAM
END
*------------------------------------------------------------------------------
MODELS
*----No turbulence models used
END
*------------------------------------------------------------------------------
BOUNDARY_CONDITIONS
*----Moving wall
  WALL         1    L    M    M    1    N
  U=UWALL V=0.0  W=0.0
*----Stationary walls.  These can be omitted, and will be imposed as default
*----   b.c.s by the pre-processor
  WALL         1    1    1    M    1    N
  U=0.0  V=0.0  W=0.0
  WALL         L    L    1    M    1    N
  U=0.0  V=0.0  W=0.0
  WALL         1    L    1    1    1    N
  U=0.0  V=0.0  W=0.0
  WALL         1    L    1    M    1    1
  U=0.0  V=0.0  W=0.0
*----Symmetry boundary
  SYMMETRY     1    L    1    M    N    N
*
END
*------------------------------------------------------------------------------
INITIAL_CONDITIONS
*----Specify velocities and pressure
  U=0.0  V=0.0  W=0.0  P=0.0
END
*------------------------------------------------------------------------------
SOLUTION_CONTROL
*----specify algorithm, solver, solution and solver iterations, and
*----indirect underrelaxation
  ALGORITHM SIMPLEC
  S_SCHEME UPWIND  U V W
  ITERATIONS 100
  SOLVER WHOLE_I U V W PP
  S_ITERATIONS 4 U V W
  S_ITERATIONS 10 PP
  INERTIAL_FACTOR 0.2  U V W
END
```

Figure 3-3. Input Data File to the Sample Problem

3.5 Output Files

The default output file from a typical SCISEAL run contains a minimum necessary amount of information. This includes SCISEAL logo, the job title, information on interface cells, and a convergence history. At the end of the file, field prints of variable values are printed if requested. Additional information such as wall information (y^+ and u^+), mass fluxes through open boundaries, etc are also printed if requested. As remarked earlier, seal specific quantities are calculated and printed in separate disk files. Additional information on run-time statistics such as linear solver convergence can be printed using a parameter described in a subsequent section. However, the default settings are recommended since the output file can grow very quickly in size and become unmanageable if additional information is requested.

SECTION 4

PARAMETERS

In the PARAMETERS section of the input file the user can define his own parameters to be used in place of numeric values in subsequent commands. The INTEGER command is used to define integer parameters and the REAL command is used to define real parameters. These commands have the following form

REAL $real_1$=$real\_expr_1$, $real_2$=$real\_expr_2$, $real_3$=$real\_expr_3$
INTEGER $integer_1$=int_expr_1, $integer_2$=int_expr_2

The expressions on the right-hand side of the assignment statements, $real\_expr_i$ and int_expr_i, can be either constant values, previously defined parameters or arithmetic expressions. Note that a comma must be used to separate a real expression from the next variable, when multiple variables are defined on the same line. A space after or before such a comma is not necessary. The following arithmetic operators can be used in expressions

Addition	+
Subtraction	-
Multiplication	*
Division	/
Exponentiation	* *
Bracket	( )

The following trigonometric and exponential functions can also be used in expressions

SIN($real\_expr_1$)	Sine
COS($real\_expr_2$)	Cosine
TAN($real\_expr_3$)	Tangent
ASIN ($real\_expr_4$)	Arcsine

ACOS (real_expr$_5$)	Arccosine
ATAN(real_expr$_6$)	Arctangent
EXP(real_expr$_7$)	Exponential

The values real_expr$_1$, real_expr$_2$, and real_expr$_3$ should be in radians. Parameter names can be up to 10 characters long and composed of alphanumeric characters. The first character must be a letter.

The names L, M, N LP1, MP1 and NP1 are reserved words and cannot be defined as parameters. The meanings of these quantities are discussed in Section 5.

Example

```
PARAMETERS
 INTEGER I1 = 10, I2 = 15, I3 = 12
 INTEGER IMAX = I1 + I2 + I3
 REAL DIST = 1.5, VEL = 20.0, VIS = 1.E-5
 REAL RE = DIST*VEL/VIS
END
```

The value of user-defined parameters can be printed by the program with the command

PRINT param$_1$ param$_2$ ...

Although integer and real constants can be used directly in command statements, it is often useful to define integer and real variables, and use them in the command lines. With the use of such "soft programming" approach, parametric testing of a problem can be done very easily by altering only the variable assignment statements, and leaving the rest of the input file unchanged.

SECTION 5

GEOMETRY

The SCISEAL grid system consists of a structured arrangement of control volumes or cells which form a computational square for two-dimensional models and a computational cube for three-dimensional models. The two-dimensional model is made up of L x M control cells where L is the number of cells in the i-direction and M is the number of cells in the j-direction. The three-dimensional model consists of L x M x N control cells where L and M are as defined for the two-dimensional model and N is the number of cells in the k-direction. The two-dimensional and three-dimensional structured meshes are shown in Figure 5-1. In each direction, there is one more cell vertex than cells.

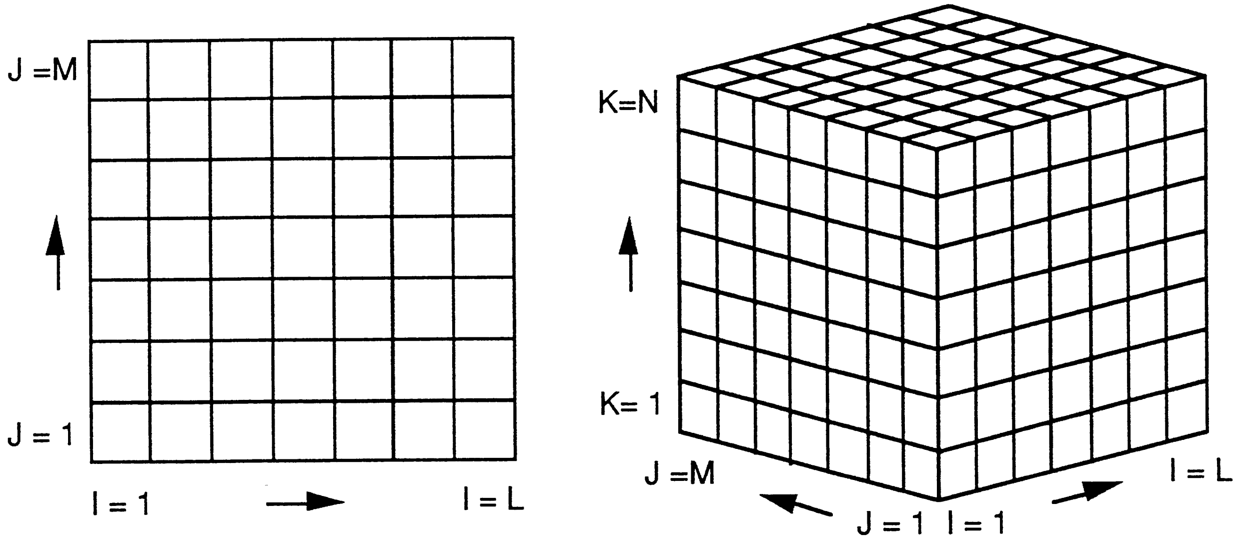


Figure 5-1. Grid Nomenclature

The grid system can be strictly cartesian or body-fitted. In the cartesian system the i-direction is aligned with the x-axis, the j-direction is aligned with the y-axis, and the k-direction is aligned with the z-axis (Figure 5-2). A body-fitted grid is one in which the i-, j-, and k-direction grid lines are allowed to bend and curve through space to conform to the boundaries of the flow domain (Figure 5-3).

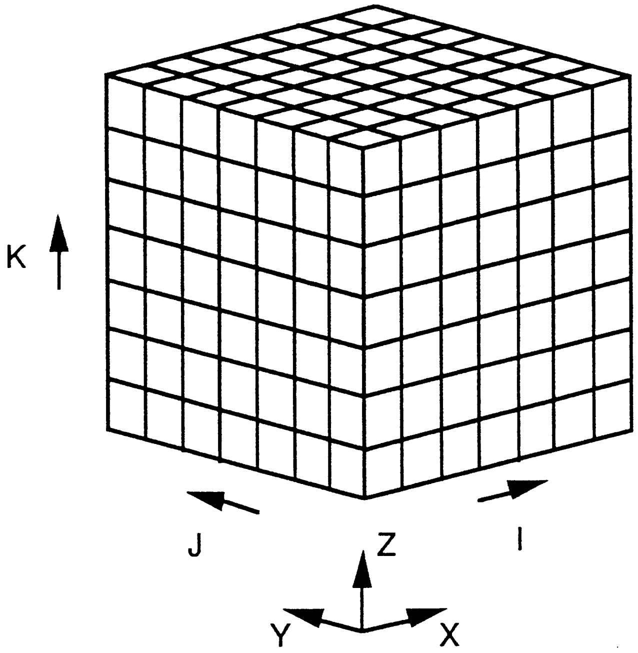


Figure 5-2. Cartesian Mesh

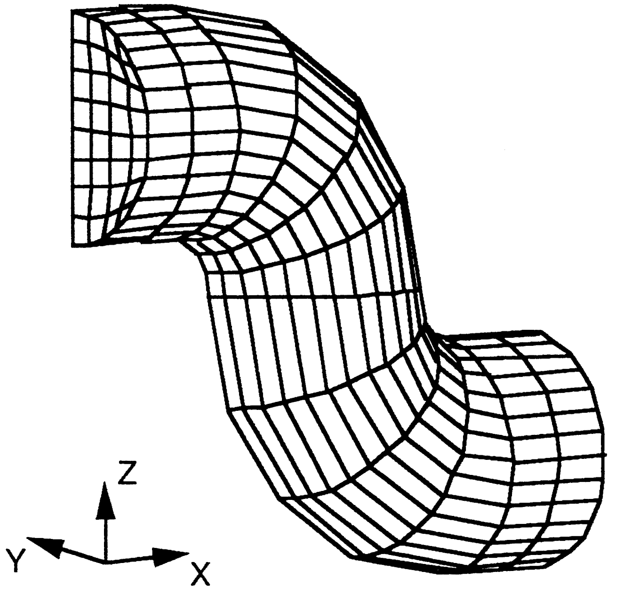


Figure 5-3. Body-Fitted Mesh

The multi-domain capability of SCISEAL allows many structured meshes to be connected to model complex geometries. The meshes must not overlap but must be joined at their boundary surfaces. The number of cells at the mesh interfaces must be the same in each of the meshes on either side of the interface. An example multi-domain mesh is shown in Figure 5-4.

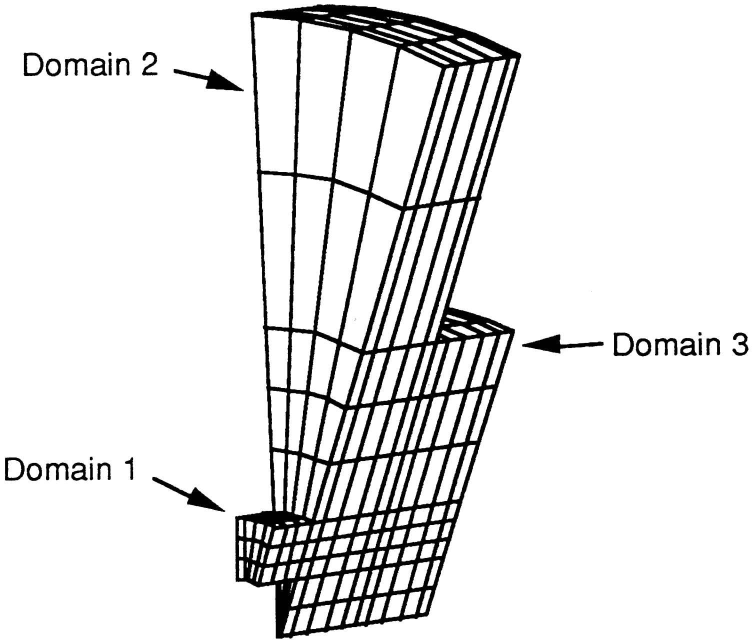


Figure 5-4. Multi-domain Mesh

The input in the Geometry section must specify the type of mesh system to be employed, the size of the mesh and the physical locations of the mesh points (cell vertices). The commands to carry out these tasks are described in the remainder of this section.

5.1 Section Format

Multi-domain models require that the user specify the number of cells and the grid point distributions for each of the domains which make up the model. This is done sequentially for each domain. When the mesh has been specified for the first domain, the DOMAIN command is entered. Then the mesh for the second domain is set followed by subsequent DOMAIN commands and mesh-setting commands as required. The GEOMETRY section would look like

```
GEOMETRY
 GRID 3D BFC
 L  L1;  M  M1;  N  N1

  Point distribution commands for domain 1


  DOMAIN 2

  L  L2;  M  M2;  N  N2

  Point distribution commands for domain 2

  etc.

 END
```

Note that the GRID command should be entered only once as the geometry type should be the same for each domain. In addition to defining the geometry of a multi-domain model, it is necessary to explicitly state how the domains are interconnected. The interconnections or interfaces are supplied in the BOUNDARY_CONDITIONS section as described in Section 9.

When the grid is read from an external source with the READ GRID command, that command should follow immediately after the GRID command. If no cell types are to be assigned, then no further input in the GEOMETRY section is required. If, however, cell types are to be assigned in the GEOMETRY section, then it is necessary to use the DOMAIN command to select the domain in which the cell types are to be set. For example, in a five-domain model, where solid cell types are to be set in the third domain only, the GEOMETRY section might look like

```
GEOMETRY
 GRID 3D BFC
 READ GRID FROM grid_file_name
 DOMAIN 2
 DOMAIN 3
 solid cell type commands
 DOMAIN 4
 DOMAIN 5
END
```

5.2 Grid Type

The first step in grid specification is to designate the type of grid system to be employed. This is done with the GRID command as follows

```
GRID [2D] [CARTESIAN          ] [AXISYMMETRIC]
     [3D] [BFC  [ORTHOGONAL]]
```

The first parameter should be either 2D or 3D which indicates the dimensionality of the model. The second parameter should be either CARTESIAN, POLAR or BFC and selects one of the coordinate systems described above. When the BFC option is selected an additional parameter can be added, the ORTHOGONAL keyword, indicating that the mesh is fully orthogonal. The AXISYMMETRIC keyword is valid for two-dimensional models only and indicates that the complete geometry is defined by an angular rotation about the x-axis. Defaults for 2-D or 3-D grid are Cartesian, orthogonal grids. The BFC grid, on specification is taken as non-orthogonal on default, unless specified as orthogonal. The 2-D grids are taken as planar, unless the axisymmetric keyword is added.

The axisymmetric option on the 2-D grid requires these to be generated in a certain manner: the axis of the axisymmetric geometry must lie along the Cartesian x-axis and the radial direction along the Cartesian y coordinate. There are no such restrictions on the direction on which a constant "i" or "j" index line lies in a BFC grid; e.g., a consent "i" line can run parallel to the Cartesian x axis.

5.3 Grid Size

The next step is to set the number of control cells to be used. This is done with the L, M, and N commands as shown below.

L	number of cells in i-direction
M	number of cells in j-direction
N	number of cells in k-direction.

Once this has been done it is necessary to set the coordinates of the grid vertices. This is described in the next sections.

Note that once L, M and N have been specified then the parameters L, M, N, LP1, MP1 and NP1 will be set by the program and can be used in subsequent commands. LP1, MP1 and NP1 are the number of cell vertices in each of the three mesh directions.

5.4 Grid Point Distribution - Cartesian Grids

Cartesian grids are relatively straightforward to specify as they require that the cell corners be specified only along a single line in each direction. All other lines use the same distribution of points. Three commands, XGRID, YGRID, and ZGRID, are used to specify the grid distribution in the x-, y-, and z- directions respectively. The commands have identical formats so detailed instructions will be provided for the XGRID command only.

A uniform distribution of L cells in the x-direction can be set up with the command

XGRID rval

where

rval is the total length of the domain in the x-direction

A more general form of the XGRID command allows setting of coordinates along a subset of the total extent of the line and allows a nonuniform grid spacing to be calculated. The general form of the command is

XGRID i_f i_l x_f x_l [distribution_options]

where

i_f is the first i-index of the line segment
i_l is the last i-index of the line segment
x_f is the starting x-coordinate of the line segment
x_l is the ending x-coordinate of the line segment

The optional distribution_options input enables the specification of nonuniform grid spacing of grid points. The format of the distribution_options varies depending upon the specific distribution option selected. This is discussed in the next section.

5.4.1 Distribution Options

Three different options for calculation of grid point spacing are available in CFD-ACE. They are the power-law formula, the hyperbolic tangent formula and the hyperbolic sine formula. The method to be used is selected with the DIST_OP command as follows

$$\textbf{DIST\_OP} \begin{bmatrix} \text{PWRLAW} \\ \text{TANH} \\ \text{SINH} \end{bmatrix}$$

The default setting is the power-law method so if no DIST_OP command is entered that method will be used. The selected method remains in effect until another DIST_OP command is encountered.

Power Law Method: The power-law method calculates grid point spacing according to the formula

$$x(i)=x_f+(x_1-x_f)*[\text{FLOAT}(i-i_f)/\ \text{FLOAT}(i_1-i_f)]{**}\text{pwr}$$

where

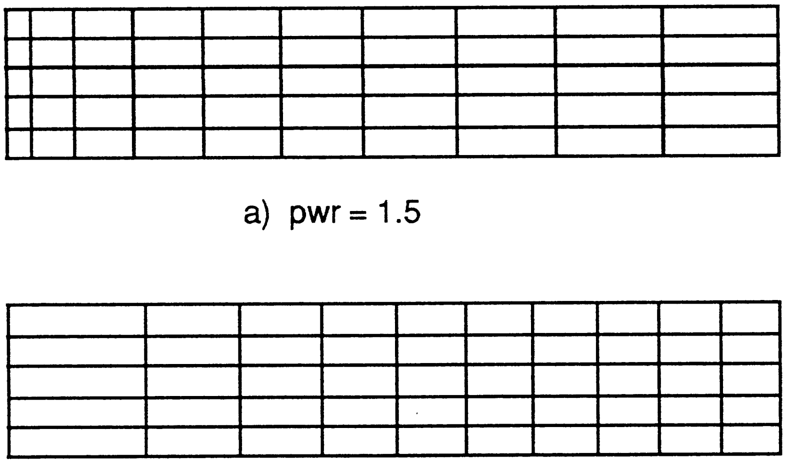
i is the index of the vertex being calculated on the line
i_f is the i-index of the first vertex on the line
i_1 is the i-index of the last vertex on the line
x_f is the x-coordinate of the first vertex on the line
x_1 is the x-coordinate of the last vertex on the line
pwr is the exponent in the above formula. Note that using pwr = 1.0 results in a uniform grid.

The distribution_options input for the power-law method has several forms which allow you to specify the power directly, to specify the beginning or ending cell width, to request a symmetric distribution of grid points and to request a forward or backward distribution of grid points. These different forms are discussed below.

Direct input of the power to be used in the power-law formula is provided with the following command

$$\text{XGRID}\ i_f\ i_1\ x_f\ x_1\ \text{pwr} \begin{bmatrix}\text{SYMM}\\ \text{B}\end{bmatrix}$$

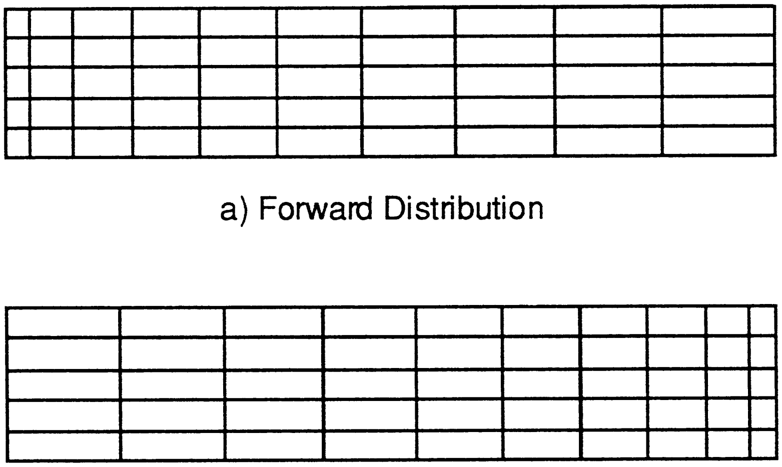
where pwr is the power. Note that a power of one results in a uniform distribution, a power greater than one will cause the grid points to be clustered toward the beginning of the line segment and a power less than one will cause the points to be clustered toward the end of the line segment (Figure 5-5). The optional parameter SYMM can be used to specify a point distribution that is symmetrical about the midpoint of the line segment. The optional parameter B can be used to specify a backward or mirror-image distribution of points (Figure 5-6).



a) pwr = 1.5

b) pwr = 0.75

Figure 5-5. Effect of Power



a) Forward Distribution

b) Backward Distribution

Figure 5-6. Forward and Backward Distributions

To specify the width of the first cell on the line segment use the command

XGRID i_f i_l x_f x_l d_f FIRST where d_f is the desired cell width. Note that the symbol '?' can be used instead of a numeric value for d_f in which case the program will set the first cell width to be the same as the immediately preceding cell width.

To specify the cell width at the end of the line, use the command

$$\text{XGRID} \;\; i_f \;\; i_l \;\; x_f \;\; x_l \;\; d_l \;\; \text{LAST}$$

where d_l is the desired cell width. Again, the symbol '?' can be used instead of an actual value in order to have the program set the cell width to match the adjacent cell width at the end of the line.

Hyperbolic Tangent Method: The calculations performed for the hyperbolic tangent method are described in Reference 3. The distribution_options required when this method is employed are

$$\text{XGRID} \;\; i_f \;\; i_l \;\; x_f \;\; x_l \;\; [\text{DF} = d_f \;\; \text{DL} = d_l]$$

where

d_f is the width of the first cell
d_l is the width of the last cell.

It is permissible to specify only the first cell width, the last cell width or both.

Hyperbolic Sine Method: The calculations performed for the hyperbolic sine method are also described in Reference 3. The distribution_options required when this option is employed are

$$\text{XGRID} \;\; i_f \;\; i_l \;\; x_f \;\; x_l \;\; \begin{bmatrix} \text{DF} = d_f \\ \text{DL} = d_l \end{bmatrix}$$

Note that either the first cell width or the last cell width can be provided but not both.

5.5 Grid Point Distribution - Boundary Fitted Coordinates

When the BFC option is used it becomes necessary to specify the x-, y- and z-coordinates of every cell vertex in the calculation domain. The general process for doing this is as follows:

- Specify the grid point coordinates along lines bounding the calculation domain or subsets of the domain.
- Generate the coordinates of points internal to the domain or sub-domain by interpolation from the points along the bounding lines.

CFD-ACE allows for points to be distributed along straight lines, arcs or splines (in the x-y plane). The commands to do this are described in the following sections.

5.5.1 Point Definition

Because the BFC grid distribution commands require specification of two- or three - dimensional coordinates it is useful to predefine physical locations using the POINT command. This is done as follows:

POINT label (x y z)

where

label is an alphanumeric character string that can be used to refer to the coordinates

x is the x-coordinate of the point

y is the y-coordinate of the point

z is the z-coordinate of the point (include for three-dimensional models only).

Any command that requires the specification of a point location can accept either the label assigned with the POINT command or the actual x-, y- and z-coordinates surrounded by parentheses.

5.5.2 Straight Lines

The following commands are used to distribute grid points on straight boundary lines for a body fitted grid.

ILINE j k i_f i_l PT_f PT_l [distribution_options]

This command is used to set up a grid distribution in the i-direction between i_f and i_l along a line of constant j and k (k should be omitted for two-dimensional models). The line is located between the two points PT_f and PT_l. The distribution_options are the same as those described for the XGRID, YGRID and ZGRID commands.

Note that the k parameter is required only for three-dimensional models. It should be omitted for two-dimensional models.

The format of the commands for the j- and k-direction grid lines are shown below

JLINE i k j_f j_l PT_f PT_l [distribution_options]
KLINE i j k_f k_l PT_f PT_l [distribution_options]

5.5.3 Arcs

Points can be distributed along arcs with the IARC, JARC and KARC commands. The IARC command is described below and illustrated in Figure 5-7.

IARC j k i_f i_l PT_c PT_f PT_l PT_n [distribution_options]

where

j	is the j-index of the line
k	is the k-index of the line (omit for two-dimensional models)
i_f	is the first i-index of the line
i_l	is the last i-index of the line

PT_c is the point at the center of the arc

PT_f is the first point on the arc

PT_l is the last point on the arc

PT_n is a point which describes the direction of the arc. The point must lie on the same side of the plane formed by PT_c, PT_f and PT_l as the end of the vector formed by rotating vector c-f into c-l in a right-handed sense.

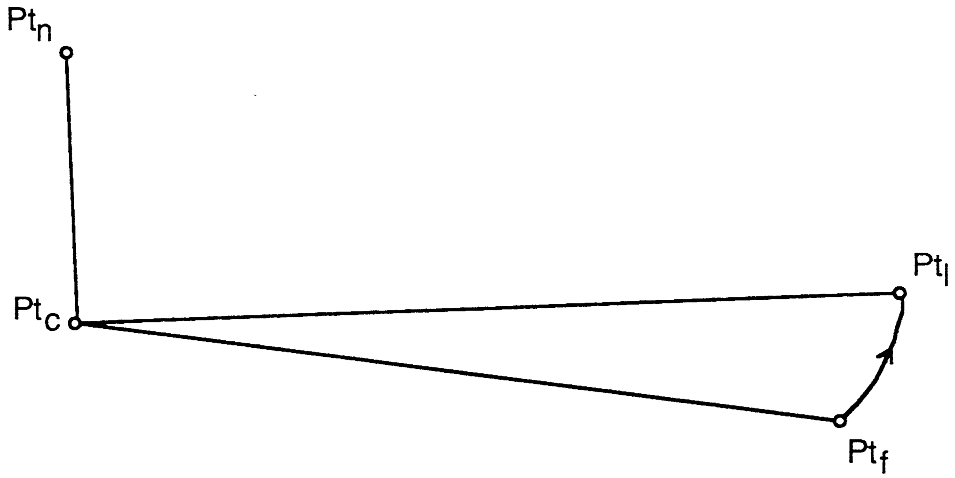


Figure 5-7. Points Used to Define an Arc

Similarly, arcs along lines in the j- and k-directions can be generated with the JARC and KARC commands illustrated below.

JARC i k j_f j_l PT_c PT_f PT_l PT_n [distribution_options]
KARC i j k_f k_l PT_c PT_f PT_l PT_n [distribution_options]

There is a slight difference in the IARC and JARC commands for two-dimensional models. The rotation vector in those cases would lie out of the x-y plane but cannot be described by a two-dimensional point which the program assumes will lie in the x-y plane. Therefore, the entry for PT_n should be replaced by '+' or '-' depending upon the sense of the rotation. It should also be noted that to be able to decide on the sense of rotation, the lines joining the center of rotation and the curve points

PT1 and PT2 can not lie in a straight line. This implies that an arc segment spanning 180° or 360° must be divided into smaller segments and generated as separate arc segments.

5.5.4 Splines

Lines that do not have a regular shape can be specified using the spline fit. The spline fit used in CFD-ACE is a third-order polynomial Akima [1] spline which will interpolate a specified set of data points. Two commands are required, the SPLINE SET command followed by the SPLINE FIT command. First, the data points are entered with the SPLINE SET command

SPLINE SET AKIMA ndp

$x_1\ x_2 \cdots x_{ndp}$

$y_1\ y_2 \cdots y_{ndp}$

where

AKIMA	Specifies the type of spline to be used.
ndp	Number of data points to be provided.
$x_1 \cdots x_{ndp}$	ndp x-coordinates
$y_1 \cdots y_{ndp}$	ndp y-coordinates

Note that there are no commas in the list of point coordinates. The coordinates are separated by blanks. If the list of coordinates is too long to fit on a single line, it can be continued to subsequent lines. The only requirement is that the list of x-coordinates and the list of y-coordinates must each begin on a new line.

The SPLINE SET command allows for three optional trailing parameters as follow:

$$\textbf{SPLINE SET}\ \text{AKIMA ndp} \begin{bmatrix} \text{ENDL} \\ \text{ENDR} \\ \text{ENDS} \end{bmatrix}$$

where

ENDL	indicates that the first two data points are to be used to set the beginning slope. The curve will start at the third point specified.
ENDR	indicates that the last two data points are to be used to set the ending slope. The curve will end at data point ndp-2.
ENDS	indicates that two additional data points are supplied at the beginning and at the end of the data points to control the slope at each end.

Note that ndp, the number of data points, must include the additional points specified for slope control at either end of the spline.

Next, the grid line along which to distribute the points and the nature of the distribution are specified with the SPLINE FIT command.

SPLINE FIT I j i_f i_l pwr [B]

where

I	indicates that the spline is to run in the mesh i-direction.
j	is the j-index of the constant j line at which the spline lies
i_f	is the i-index of the first point on the line
i_l	is the i-index of the last point on the line
pwr	is the exponent in the power law formula according to which the grid spacing is determined.
[B]	is an option to distribute the points backward from i_l to i_f. By default, the distribution is made forward from i_f to i_l.

The form of the SPLINE FIT command along a line of constant i is

SPLINE FIT J i j_f j_l pwr [B]

5.6 Distribution of Interior Points

After the boundary points are located, the interior points can be calculated by interpolation from the boundary points. To build the mesh over a computational plane use the command

FILSRF i_f i_l j_f j_l k_f k_l

where

i_f is the i-direction index of the first point in the region
i_l is the i-direction index of the last point in the region
j_f is the j-direction index of the first point in the region
j_l is the j-direction index of the last point in the region
k_f is the k-direction index of the first point in the region (3D only)
k_l is the k-direction index of the last point in the region (3D only)

To construct the mesh over a computational volume use the command

FILVOL i_f i_l j_f j_l k_f k_l

The parameters are the same as for the FILSRF command.

It is only necessary to distribute points along the boundary lines of the volume. Points on the interior of the surfaces and of the volume will be calculated automatically.

It is sometimes necessary to split up a bounding domain surface into several smaller pieces and generate separate surface grids on these to preserve local domain details. In such a case, the default FILVOL command cannot be used, since the FILVOL command will overwrite the details needed in the sub-sections of the volume faces. The proper method to generate such a grid is then to individually generate surface grids on all the smaller pieces of the bounding face using the FILSRF command for eah smaller piece. Then a modified FILVOL command should be used to preserve the details on the bounding faces:

FILVOL i_f i_l j_f j_l k_f k_l NO

When using the NO qualifier with FILVOL, note that all the bounding surface must have the surface grids generated using the FILSRF command. If any of these have not been previously generated the resulting grid will not be usable.

5.6.1 Projection of 2-D Mesh to Create a 3-D Mesh

Three-dimensional volume meshes can be created by linear projection of two-dimensional surface meshes with the following commands

PROJECT_I i_f i_l j_f j_l k_f k_l dx dy dz pwr [SYMM] [B]
PROJECT_J i_f i_l j_f j_l k_f k_l dx dy dz pwr [SYMM] [B]
PROJECT_K i_f i_l j_f j_l k_f k_l dx dy dz pwr [SYMM] [B]

where

i_f is the first i-index of the three-dimensional mesh
i_l is the last i-index of the three-dimensional mesh
j_f is the first j-index of the three-dimensional mesh
j_l is the last j-index of the three-dimensional mesh
k_f is the first k-index of the three-dimensional mesh
k_l is the last k-index of the three-dimensional mesh
dx is the x-component of the projection vector
dy is the y-component of the projection vector
dz is the z-component of the projection vector
pwr is the power in the power-law which determines the distribution of mesh planes along the projection vector

Note that the PROJECT_I command operates on the mesh at $I=i_f$. It replicates the mesh and translates it an appropriate distance at each i from i_f +1 to i_l. The PROJECT_J and PROJECT_K commands operate in a similar fashion to allow projection of mesh planes in each of the grid directions. It should be noted here that the index I refers only to the grid and not to the physical coordinate directions. Thus, a 2-D mesh can be projected parallel to the cartesian Z axis, but with

PROJECT_I command.

5.6.2 Revolution of 2-D Mesh to Create a 3-D Mesh

Three-dimensional volume meshes can be created by revolution of a two-dimensional mesh about a specified axis with the following commands

REVOLVE_I i_f i_l j_f j_l k_f k_l Pt_1 Pt_2 alpha pwr [SYMM] [B]
REVOLVE_J i_f i_l j_f j_l k_f k_l Pt_1 Pt_2 alpha pwr [SYMM] [B]
REVOLVE_K i_f i_l j_f j_l k_f k_l Pt_1 Pt_2 alpha pwr [SYMM] [B]

where

i_f is the first i-index of the three-dimensional mesh
i_l is the last i-index of the three-dimensional mesh
j_f is the first j-index of the three-dimensional mesh
j_l is the last j-index of the three-dimensional mesh
k_f is the first k-index of the three-dimensional mesh
k_l is the last k-index of the three-dimensional mesh
Pt_1 is the first point on the axis of rotation
Pt_2 is the second point on the axis of rotation
alpha is the angle of rotation in degrees
pwr is the power in the power-law which determines the angular distribution of mesh planes

The indices I, J, or K in REVOLVE command refer to the grid index. Thus, a REVOLVE_I command generates mesh plates along 'i' index direction, and the rotational axis is defined by the two points PT1 and PT2.

5.6.3 Translation of Mesh Regions

A region of the mesh can be moved in its entirety to another location in space with the following command

TRANSLATE i_f i_l j_f j_l k_f k_l dx dy dz

where

i_f is the first i-index of the mesh region
i_l is the last i-index of the mesh region
j_f is the first j-index of the mesh region
j_l is the last j-index of the mesh region
k_f is the first k-index of the mesh region
k_l is the last k-index of the mesh region
dx is the x-component of the translation vector
dy is the y-component of the translation vector
dz is the z-component of the translation vector

5.6.4 Rotation of Mesh Regions

A region of the mesh can be rotated in its entirety about the Cartesian x-, y-, or z-axis with the following commands

ROTATE_X i_f i_l j_f j_l k_f k_l alpha
ROTATE_Y i_f i_l j_f j_l k_f k_l alpha
ROTATE_Z i_f i_l j_f j_l k_f k_l alpha

where

i_f is the first i-index of the mesh region
i_l is the last i-index of the mesh region
j_f is the first j-index of the mesh region
j_l is the last j-index of the mesh region
k_f is the first k-index of the mesh region
k_l is the last k-index of the mesh region
alpha is the rotation angle in degrees

5.7 Reading An Externally Created Grid

Grid point coordinates can be read from an external file using the command

READ GRID FROM filename $\begin{bmatrix}\text{UNFORMATTED}\\\text{FORMATTED}\end{bmatrix}$

Both formatted and unformatted grid files can be read, the choice being made by the last keyword. If the keyword is omitted, the default file type is formatted. Details of the grid file format are provided in Appendix A.

If the READ GRID command is used, the program will make a new copy of the grid file for the analysis program to read. If the grid file to be read requires no change, such as multiplication by a scaling factor, then it can be read directly by the analysis program. This option is selected by the command

USE GRID FROM filename $\begin{bmatrix}\text{UNFORMATTED}\\\text{FORMATTED}\end{bmatrix}$

5.8 Mesh Scaling

A scale factor can be applied to the mesh with the following command

SCALE factor

All mesh coordinates will be multiplied by the specified factor. Note that mesh scaling is not performed until the mesh data is written to disk immediately before the preprocessor terminates. Therefore, any input commands which require geometric data should use the original mesh coordinates and not the scaled coordinates.

5.9 Geometric Tolerance

When multi-block grids are employed, the preprocessor must determine the interface boundaries which map to each other. This is done by checking for physical coincidence of grid points at interface boundaries. The tolerance used in this calculation can be automatically determined by the program or can be input by the user with the command

TOLERANCE dtol

It is best to first let the program determine the tolerance itself. If any unmatched interfaces are found, the program will write a warning message. If this happens, it may be that the tolerance calculated by the program was too small. In this case, providing a tolerance of about 10 times what the program initially used may allow the interface checking to succeed. If it still fails, it will be necessary to check the grid to ensure that domain boundaries are indeed coincident and that the interface boundaries are properly specified.

5.10 Blockage

Regions of the grid can be blocked to flow using the BLOCK command.

BLOCK i_f i_l j_f j_l k_f k_l

where

i_f is the i-direction index of the first cell in the region
i_l is the i-direction index of the last cell in the region
j_f is the j-direction index of the first cell in the region
j_l is the j-direction index of the last cell in the region
k_f is the k-direction index of the first cell in the region (3D only)
k_l is the k-direction index of the last cell in the region (3D only)

Blocked cells represent an obstruction inside the flow domain where flow as well as heat conduction is not allowed. Existence of internal solid objects can make a computational grid complicated and sometimes difficult to generate. Use of blocked cells can facilitate better quality grids and easy grid generation in such cases.

5.11 Solid Regions

Solid regions are regions of the mesh which represent heat-conducting solid material. By meshing into a solid region, conjugate heat transfer calculations can be performed. Solid regions are defined using the SOLID command,

SOLID imat i_f i_l j_f j_l k_f k_l

where

imat is a character string which will be used in the PROPERTIES section to assign thermal properties to the solid region

and the remaining parameters identify the region of the mesh in the same manner as the BLOCK command.

The SOLID and BLOCK commands both represent a solid part of a flow domain, and are treated in the same fashion when dealing with all variables except the energy equation. With SOLID specification, the solid part is also treated as a part of the domain for the energy equation, while a blocked region is ignored.

5.12 Seal Grid Generation

The present version of SCISEAL can be set up to treat a large variety of seals, but has been specifically geared to treat cylindrical seals (annular, axial stepup, axial stepdown, and temporal land seal). These include easy grid generation, boundary condition specification, and flow and rotordynamic coefficient computation. The geometries of cylindrical seals are relatively easy to generate using a small number of parameters. Several standard cylindrical seal types have been included in the preprocessor, and the geometry as well as the grid can be generated using special statements in the input files.

With the capabilities available in SCISEAL, it is possible to set up a generic seal geometry in several different orientations. However, to simplify the treatment of all the seal-related pre- and post-processing functions and grid generation commands, some restrictions are needed on the geometry orientation and the placement of the boundary conditions. It should be noted that correct flow solutions results can be obtained in any other compatible geometrical configuration, however, the post-processing routine which calculates the integrated loads, dynamic coefficients *etc*. will then provide erroneous results.

The generic cylindrical seal consists of the rotor and the stator, as shown in Figure 5-8. The axis of the rotor is assumed to lie along the X axis of the Cartesian reference frame, and the cross-sectional planes lie in the Y-Z plane of the reference frame. The Y axis coincides with a rotor radius, which is also taken to be the direction along which the rotor is displaced for calculations with static eccentricity (steady solutions as well as circular whirl orbit) is introduced. This is also the line of motion of the rotor center when the moving grid formulation is used to calculate the rotordynamic coefficients.

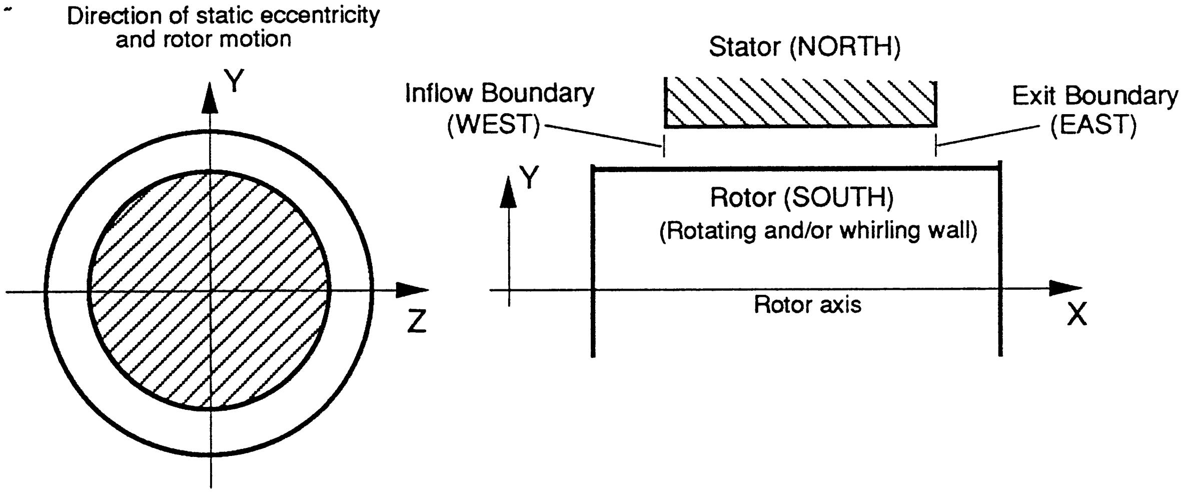


Figure 5-8. Details of a Generic Cylindrical Seal Configuration

Four type of seals with cylindrical geometry are described in the following sub-sections. Each of the seals is identified by a number, and the relevant grid and geometry specifications are provided.

NOTE: To simplify the form of the **GCSEAL** command, the maximum number of the cells in the three directions, L, M, and N are assumed known. Hence, to assure grid accuracy, the **GCSEAL** command must be placed after the values of the indices

L, M, and N have been set in the GEOMETRY section.

5.12.1 Annular Seal

This is the simplest of the four seal types considered here. The relevant parameters are:

Geometry

- R_R : Radius of the rotor
- R_S : Radius of the stator
- X_L : Axial length of the seal
- C_O : Nominal radial clearance, R_S - R_R
- e : Eccentricity ratio, ε/C_O

A section along a seal length is shown in Figure 5-9 to illustrate these parameters.

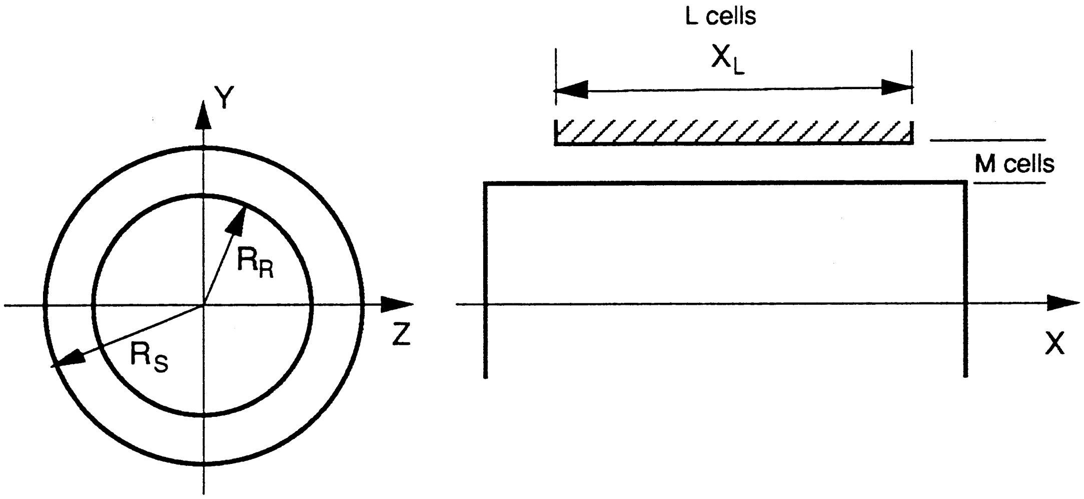


Figure 5-9. Geometry of the Annular Seal

A grid for this type of seal can be generated by using the command:

GCSEAL 1 X_L R_R R_S R_{Y1} e θ_Y θ_Z

Where $\mathbf{R_{Y1}}$ is the grid clustering parameter in the radial direction. The grid clustering is done symmetrically to provide smaller spacing near both the walls. The default values of $\mathbf{e}$ and $\mathbf{R_{Y1}}$ are 0.0 and 1.0 respectively. θ_Y and θ_Z are the shaft misalignment angles in the Y and Z directions respectively. For a proper grid generation, the Y axis must be aligned with the Y axis, and then the misalignment should be specified. Please note that the misalignment effects are ignored when the grid generation command is used to calculate seal rotordynamic coefficients.

5.12.2 Axial Step-Down Seal

This seal contains a contractive step in the flow direction. This step is assumed to be on the stator wall, *i.e.* the rotor is assumed to be cylindrical. A cross section of the seal is shown in Figure 5-10. The relevant parameters for this seal are:

Geometry

$\mathbf{R_R}$: Radius of the rotor

$\mathbf{R_{SI}}$: Radius of the stator after the contractive step

$\mathbf{R_{SO}}$: Radius of the stator before the contractive step

$\mathbf{X_{L1}}$: Axial length of the seal before the step

$\mathbf{X_{L2}}$: Axial length of the seals after the step

$\mathbf{C_O}$: Nominal minimum radial clearance $\mathbf{R_{SI}}$ - $\mathbf{R_R}$

$\mathbf{e}$: Eccentricity ratio, ε/C_O

Grid

$\mathbf{L_1}$: Number of cells in the axial (ξ) direction before the step

$\mathbf{L_2}$: Number of cells in the axial (ξ) direction after the step

$\mathbf{M_1}$: Number of cells in the radial (η) direction between $\mathbf{R_R}$ and $\mathbf{R_{SI}}$.

$\mathbf{M_1}$: Number of cells in the radial (η) direction between $\mathbf{R_R}$ and $\mathbf{R_{SO}}$.

The grid for this geometry can be generated using the following command:

GCSEAL 2 L_1 L_2 M_1 M_2 X_{L1} X_{L2} R_R R_{SI} R_{SO} R_{X1} R_{Y1} R_{Y2} e

Where R_{X1}, R_{Y1}, and R_{Y2} are the clustering parameters. R_{X1} generates a non-uniform grid with clustering near the radial wall at the axial step. R_{Y1} generates a symmetrically clustered grid in the gap between the rotor and the smaller stator radius. R_{Y2} is the clustering parameters for symmetrical clustering of grids between the inner and outer radii of the stator. An index check in also performed to see whether L_1 and L_2 add up to L and M_2 equals M.

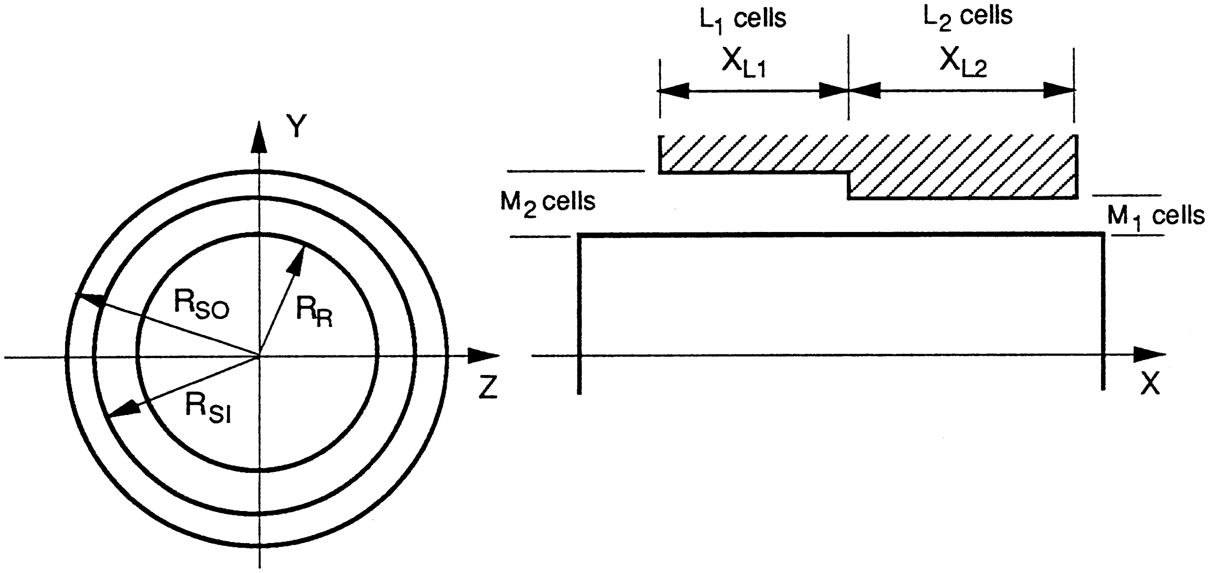


Figure 5-10. Geometry of the Axial Step-Down Seal

5.12.3 Axial Step-Up Seal

This seal is similar to the previous seal, except that an expansion step is present in the axial flow direction. The grid parameters are (Figure 5-11) :

Geometry

R_R : Radius of the rotor

R_{SI} : Radius of the stator before the expansion step

R_{SO} : Radius of the stator after the expansion step

X_{L1} : Axial length of the seal before the step

X_{L2} : Axial length of the seals after the step

C_O : Nominal minimum radial clearance, R_{SI} - R_R

e : Eccentricity ratio, ε/C_O

Grid

L_1 : Number of cells in the axial (ξ) direction before the step

L_2 : Number of cells in the axial (ξ) direction after the step

M_1 : Number of cells in the radial (η) direction between R_R and R_{SI}.

M_1 : Number of cells in the radial (η) direction between R_R and R_{SO}.

The grid for this geometry can be generated using the following command:

GCSEAL 3 L_1 L_2 M_1 M_2 X_{L1} X_{L2} R_R R_{SI} R_{SO} R_{X1} R_{Y1} R_{Y2} e

The clustering parameter R_{X1} now generates a non-uniform grid in the axial direction after the expansion step with packing near the wall. The clustering parameters in the Y direction have meaning similar to that in the previous case. An index consistency check is done on L_1, L_2, and M_2 as in the axial step-down seal case.

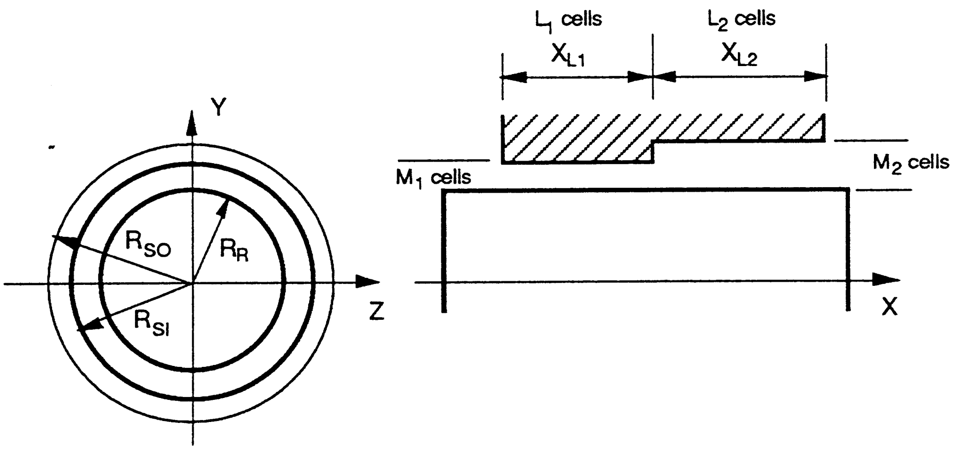


Figure 5-11. Geometry of the Axial Step-Up Seal

5.12.4 Tapered Land Seal

This type of seal has a convergent tapered portion along a part or all the length of the seal in the flow direction. A section along the seal length is shown in Figure 5-12, and the geometry parameters are:

Geometry

R_R : Radius of the rotor

R_{SO} : Radius of the stator at the beginning of convergent portion

R_{SI} : Radius of the stator in the straight portion

X_{L1} : Axial length of the tapered portion

X_{L2} : Axial length of the straight portion

C_O : Nominal minimum radial clearance, R_{SI} - R_R

e : Eccentricity ratio, ε/C_O

Grid

L_1 : Number of cells in the axial (ξ) direction in the tapered portion.

L_2 : Number of cells in the axial (ξ) direction in the straight portion.

The grid for this geometry can be generated using the following command:

GCSEAL 4 L_1 L_2 X_{L1} X_{L2} R_R R_{SI} R_{SO} R_{Y1} e

To generate a seal without a straight portion, the number of cells in the straight portion, L_2, should be set to zero. When L_2 is zero, the length X_{L2} will be ignored. An index check in performed on L_1 and L_2 to see whether they add up to **L**.

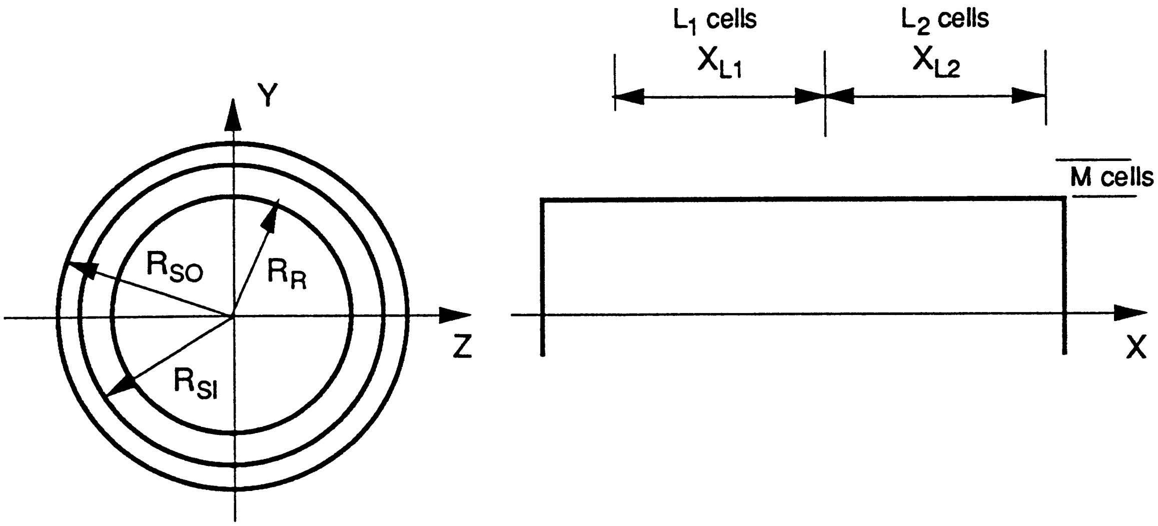


Figure 5-12. Geometry of the Tapered Land Seal

5.13 External Grid with Rotordynamics

When solving for the rotordynamic coefficients of an annular seal, the SCISEAL code has been geared for certain grid and index arrangements. For this reason, it is recommended that the GCSEAL commands in single-domain problems be used. However, for more complicated problems it is still possible to use an externally created grid which can be read in the pre-processor. Certain rules must be observed to successfully obtain solutions for rotordynamic coefficients (see Section 6.3). Following are guidelines for setting up the grids and boundary conditions in the two rotordynamic coefficient calculation methods.

5.13.1 Whirling Rotor Method

The 3-D grid should be generated such that the stator centerline coincides with the x-axis. The rotor should be placed in an eccentric position along the positive y-axis (see Figure 5-8). The eccentricity ratio should be small to maintain accuracy of the rotordynamic coefficient method and a 10% eccentricity ratio (ε/C_o), is recommended where ε is the static eccentricity and C_o is the nominal (minimum) seal clearance.

Note also that the full 360° portion of the seal must be used in the problem definition. The 'i' index should be assigned to lines running along the axial direction so that the seal inlets and outlets are on west or east faces. The grid lines along the radial direction should be along the j index so that the 'south' boundary represents the rotor surface, with imposed spin. North boundary is the stator wall. The grid lines running in the tangential direction are assigned the k index and the 'low' and 'high' surfaces are periodic boundaries and are specified using the cyclic k command (see Section 9.4). Care should be taken during the assignment of the i,j,k directions to maintain a right-handed coordinate system, otherwise the code will generate negative volumes and terminate execution.

5.13.2 Perturbation Method

The perturbation method uses 'small' perturbation about the nominal rotor center to calculate the rotordynamic coefficients. The grid for this method should have the same topology as described for the whirling rotor method (i.e., the grid index arrangement), with the various boundary conditions specified as outlined above. One difference is that now the rotor centerline should coincide with the cartesian x-axis. The rotor can be at a nominally centered position or can have a non-zero static eccentricity. Since the perturbation model imposes 'small' rotor motion internally, there is no restriction on the static eccentricity ratio. In this case the stator centerline should be shifted in the negative y-direction by an amount equal to the static eccentricity so as to maintain the rotor centerline at cartesian x-axis. The perturbation model relies on small imposed whirl on the rotor and to simplify the model additional constraints are required on the grid. The radial grid lines, i.e. lines along the 'j' index must lie at the same axial distance. In other words, the constant 'i' grid planes should traverse the complete seal gap. The model will allow blocked cells on rotor wall or stator wall or both, but a partial block in the 'y' direction that does not join the south or north walls is not allowed. Partial blocks in the 'i' or 'k' directions, however, can be handled by the perturbation model.

SECTION 6

PROBLEM TYPE

The PROBLEM_TYPE input section is used to specify the type of problem to be solved and to state whether the problem is steady state or time-dependent.

6.1 Variables To Be Solved

The SOLVE command is used to specify the variables to be solved.

$$\text{SOLVE}\begin{bmatrix}\text{FLOW}\\ \text{FLOW\_C}\end{bmatrix}\text{SWIRL TURBULENCE HEAT MIXING}$$

The parameters have the following meanings

FLOW	incompressible flow
FLOW_C	compressible flow
SWIRL	θ-direction velocity component in two-dimensional and AXISYMMETRIC models
TURBULENCE	turbulent flow
HEAT	heat transfer
MIXING	mixing of two or more 'compositions' of chemical species

Note that the HEAT parameter alone can be used to solve a conduction problem, but that all other parameters must be used in conjunction with the FLOW or FLOW_C parameters. SWIRL option is valid for 2-D axisymmetric flows only. The MIXING option can be used to solve for the mixing of one or more streams of fluids with different specified composition. Usually, mixing problems involve heat transfer, although the SCISEAL code will allow the MIXING parameter without HEAT parameter.

The problem type has no default option, to provide the flexibility of solving for heat conduction problems alone. Use of FLOW or FLOW_C indicates a laminar incompressible or compressible flow. Subsequent keywords then add additional solution variables. The absence of the SOLVE section will prompt an error message from the pre-processor.

More than one SOLVE statement can be entered. The effect of multiple SOLVE statements is cumulative, as if all the arguments of all statements had been included in a single statement.

6.2 Time Dependence

By default, the CFD-ACE program solves for steady state flow conditions. For transient calculations, the following command is required

UNSTEADY TF=t_f TL=t_l STEPS=nsteps

where the parameters t_f, t_l and nsteps have the following meanings

t_f	is the start time
t_l	is the end time
nsteps	is the number of time steps

6.3 Rotordynamic Coefficients

Rotordynamic coefficients relate the changes in the fluid reaction forces developed by flow in the seal as a result of small perturbations in the position of the rotor as it spins in the seal. The rotor may move to produce only eccentricity, or misalignment, or both. At present SCISEAL is set up to compute the fluid forces when the rotor movement produces only eccentricity. *i.e.* the rotor and stator axes remain parallel. For a nominally centered rotor position, and when the rotor motion is small, the reaction forces can be linked to the rotor axis displacement, velocity, and acceleration as (see Figure 5-8):

$$-\begin{bmatrix} F_y \\ F_z \end{bmatrix} = \begin{bmatrix} K & k \\ -k & k \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} C & c \\ -c & C \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} M & o \\ o & M \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{z} \end{bmatrix}$$

where **K** and **k** are the direct and cross-coupled stiffness coefficients, **C** and **c** are the direct and cross-coupled damping coefficients, and **M** is the lumped mass parameter.

At present there are two options to calculate the rotordynamic coefficients for the centered seal geometry. These are:

1. **Circular whirl orbit method:** In this method, a circular whirl orbit is imposed on the rotor center from its **centered** position. A coordinate transformation is made to make the flow quasi-steady. The flow solutions are calculated at several whirl frequencies, and the fluid pressures are integrated on the rotor surfaces to provide reaction forces at these whirl frequencies. A curve fit is then used to calculated the rotordynamic coefficients. This method is set up to calculate the symmetric coefficient matrix associated with the centered rotor.

2. **Small perturbation method**: When the rotor is at a statically eccentric and/or misaligned position, the perturbation method must be used. This method uses a small time-dependent perturbation in the rotor position and the corresponding time-dependent flow changes are calculated. The method employs complex quantities to represent the perturbation in the flow variables. These are solved for using complex algebra. The reaction force-displacement relation used in this method is:

$$-\begin{bmatrix} F_y \\ F_z \end{bmatrix} = \begin{bmatrix} K_{yy} & K_{yz} \\ -K_{zy} & K_{zz} \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} C_{yy} & C_{yz} \\ -C_{zy} & C_{zz} \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} M_{yy} & M_{yz} \\ -M_{zy} & M_{zz} \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{z} \end{bmatrix}$$

where F_y F_z are the fluid reaction forces which are linked to the rotor displacement, velocity, and acceleration (y,z, $\dot{y}, \dot{z}, \ddot{y}, \ddot{z}$) through the stiffness, damping, and inertia coefficients (K, C, M).

displacement, velocity, and acceleration (y,z, $\dot{y}, \dot{z}, \ddot{y}, \ddot{z}$) through the stiffness, damping, and inertia coefficients (K, C, M).

The perturbation method provides all twelve coefficients. This equation reduces to that given earlier for centered rotor position.

The calculation of rotordynamic coefficients for cylindrical seals can be performed using the following command

ROTDYN $\begin{bmatrix}\text{TRANSF}\\ \text{PERTB}\end{bmatrix}$ Omega C_o ECNRAT

where

TRANSF : whirling rotor method for nominally concentric rotor, generates skew-symmetric coefficient matrices

PERTB : N-S perturbation method for both concentric and eccentric rotors, will generate full coefficient matrices

Omega : rotational speed of the rotor in rpm

C_o : nominal seal clearance, meters

ECNRAT : eccentricity ratio for the TRANSF method

As outlined earlier, the whirling rotor method needs a computational grid where the rotor is whirling in a small circular orbit. This is accomplished by generating a grid where the rotor has non-zero static eccentricity. When the GCSEAL command is used, ECNRAT is set to 0.1, and the specified value here will be ignored. But when an external grid is used, ECNRAT must be specified since it is needed for the rotordynamics calculations. A value of 0.1 is recommended and guidelines on generating the grid were outlined in Section 5.12.

Several points should be noted concerning the coefficient calculation procedure:

1. The whirling rotor method is applicable to nominally centered rotor. If the GCSEAL command is used to generate the grid then the code will internally generate grids during the rotordynamics calculations. The whirling rotor needs a seal configuration where the rotor is whirling in a small circular orbit about the stator center. This is achieved by re-generating the grid with a slight static eccentricity of the rotor. During this phase, static eccentricity, if any, is imposed in the GCSEAL command will be ignored.

2. Externally created grids can also be used together with the whirling rotor method. In this case, the code will skip the non-whirling solution. However, the first whirl frequency corresponds to the non-whirl case since the whirl frequency is set to zero. The grid generation for this case must be done following certain rules which were outlined in Section 5.13.

3. The perturbation method is designed to work with the rotor axis always coinciding with the cartesian x-axis. If the internal grid generation procedure (GCSEAL commands) are used to generate the grid, this condition is automatically imposed. In addition, all rotating walls are assumed as a part of the rotor and the grid line indexing is done so that the 'j' index runs across the seal gap and these grid lines are along the radial directions, i.e., the lines at a given axial location form a flat surface which is perpendicular to the rotor axis.

4. The perturbation method can also be used on an externally created grid. Several species rules must be followed that are described in Section 5.13. The rotor in this case can be statically eccentric with respect to the stator. However, the rotor centerline must still coincide with the cartesian x-axis.

5. When using externally created grids with any of the rotordynamic methods, care must be taken to arrange the grid line indices as described in Section 5.13. In addition, the boundary conditions <u>must</u> be specified in a certain fashion for the calculation procedure to be

successful. In brief, the seal inlet and outlet planes must lie along west or east boundaries. The stator wall(s) must lie on the north boundaries while the south boundary should be assigned to all spinning rotor suface(s). The low and high boundaries must be specified through the cyclic k command. These points were explained in Section 5.13, and are again described here to remind the user the proper way of the grid generation procedure and boundary conditions specification that will ensure correct rotordynamic coefficients.

SECTION 7

PROPERTIES

The PROPERTIES input section is used to set the properties of the fluid involved in the simulation. The relevant properties will depend upon the type of flow simulation requested in the PROBLEM_TYPE section. That is, an incompressible flow problem requires that density and viscosity be specified. When heat transfer is added, the fluid specific heat and Prandtl number must also be given.

SCISEAL contains a data base of properties for a large number of gaseous species. For single-fluid problems, the FLUID command can be used to specify the constituents of the fluid and the program will automatically calculate all the relevant properties. However, if none of the built-in fluids is appropriate, the fluid properties, viscosity, density, and specific heat can be input directly. Similarly, for mixing flow problems, the COMPOSITION command is used to specify the constituents of two or more fluids and the program will use the built-in property data to calculate mixture properties.

7.1 Compositions

Flows involving mixing of two of more gaseous components requires the specification of the chemical species of interest. This input is provided by the COMPOSITION command

COMPOSITION icmps sp_1 m_{sp1} sp_2 m_{sp2} ...

where

icmps	is the composition number
sp_i	is the species name such as CO_2 or H_2O
m_{spi}	is the mass fraction of sp_i.

The COMPOSITION statement has to be specified for each composition entering the calculation domain. The quantity ($m_{sp1}+m_{sp2}+...+m_{spn}$) for each composition has to equal unity. The number of equations (for the compositions) that will be solved is equal to 1 less than the total number of compositions specified. The order of the composition definition is not important.

The species database includes the following twenty-five species

CH_3, CH_4, C_2H_4, C_3H_6, C_3H_8, C_7H_{16}, $C_{10}H_{19}$, CH_6N_2, CO, CO_2, H_2, O_2, N_2, H_20, HCN, OH, H, O, N, NO, NO_2, N_2O_4, N_2H_4, AR, HE

The user can add up to ten species to the program database by providing the necessary data in a file. The file can then be read by the program with the command

THERMO_DATA FROM filename

The data that must be provided and the format of the data file are described in Appendix B.

7.2 Density

The DENSITY command is used to specify the way in which fluid density is determined by the program. It has the form

$$\textbf{DENSITY} \begin{bmatrix} \text{CONSTANT} & \text{rval} & \\ \text{INVERSE\_T} & \text{PRESS} = p_{ref} & \text{MOL\_WT} = w_m \\ \text{GAS\_LAW} & \text{PRESS} = p_{ref} & \text{MOL\_WT} = w_m \end{bmatrix}$$

where the meanings of the parameters are described below.

CONSTANT Density will be taken as a constant value, rval.

INVERSE_T Density will be calculated according to the ideal gas formula

$$\rho = \frac{p_{ref}{}^{*}w_m}{GASCON * T}$$

where p_{ref} is a reference pressure and w_m is the local molecular weight of the fluid, T is the local fluid absolute temperature, and GASCON is the universal gas constant (8314 J/kmol · K).

GAS_LAW — Density will also be calculated according to the ideal gas formula, but it will be dependent on the local pressure as

$$\rho = \frac{\left(p + p_{ref}\right)^{*}w_m}{GASCON * T}$$

where p is the relative local pressure calculated by the program.

In the second two options, P_{ref} is a reference pressure that will be added to the calculated pressure field. It is required that P_{ref} be provided for the INVERSE_T option but not for the GAS_LAW option. Also, in those two options, w_m is the fluid molecular weight. It is not required if the FLUID command has been used or if COMPOSITIONS have been specified.

The default density option is CONSTANT.

When solving a natural convection problem with gravity sources, a reference density is needed for the body force calculations. This value typically corresponds to a suitable mean value or free-stream/quiescent conditions. The reference density value is specified using the command

```
RHOREF rval
```

If not specified, the pre-processor will try to decide on the value to be used based on the boundary conditions. However, this value may not be the desired one, and it is recommended that a correct value be assigned explicitly.

7.3 Viscosity

7.3.1 Newtonian Fluids

The fluid viscosity used by the program is determined by the VISCOSITY command.

$$\text{VISCOSITY} \begin{bmatrix} \text{CONSTANT\_DYNAMIC} & \text{rval} \\ \text{CONSTANT\_KINEMATIC} & \text{rval} \\ \text{SUTHERLAND}\ \ A = \text{rval}_a & B = \text{rval}_b \\ \text{POLY5}\ c_0\ c_1\ c_2\ c_3\ c_4\ c_5 & \\ \text{MIX\_SUTHERLAND} & \\ \text{MIX\_POLY5} & \end{bmatrix}$$

A constant value of either dynamic of kinematic viscosity can be specified using either of the first two options. When the keyword SUTHERLAND is used the program will calculate the dynamic viscosity at each cell according to Sutherland's law. The keyword POLY5 indicates that the viscosity will be calculated according to a fifth order polynomial in temperature. Coefficients must also be provided as described below. The keyword MIX_SUTHERLAND indicates that the viscosity is to be calculated from local composition and temperature. The keyword MIX_POLY5 indicates that the viscosity will be calculated according to a fifth-order polynomial in temperature for each species and the fluid viscosity at each control cell will be dependent on the local species composition.

Sutherland's law is typically written as

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + S_1}{T + S_1}$$

where T_0 is a reference temperature, μ_0 is the viscosity at the reference temperature and S_1 is a constant. This formula can be rearranged to be expressed in the following form

$$\mu = \frac{AT^{3/2}}{(B+T)}$$

and it is the coefficients A and B that must be entered.

When the POLY5 option is selected, then the fluid viscosity will be calculated as a fifth-order polynomial in temperature, i.e.

$$\mu = c_o + c_1 T + c_2 T^2 + c_3 T^3 + c_4 T^4 + c_5 T^5$$

When the MIX_SUTHERLAND option is selected (for MIXING problems) the fluid viscosity will be calculated using the built-in Sutherland's law coefficients for each species in the system and the local species concentrations.

When the MIX_POLY5 option is selected (for MIXING problems) the fluid viscosity will be calculated according to the viscosity of individual species and local species concentrations. Viscosities of individual species will be calculated at fifth order polynomial functions of temperatures. The polynomial coefficients for each species must be input with the command

VISC_POLY5_COEFS species_name c_0 c_1 c_2 c_3 c_4 c_5

One such command must be entered for each species in the system.

The default viscosity option is CONSTANT unless the flow involves mixing or reaction in which case the MIX_SUTHERLAND option is selected.

7.4 Specific Heat

For flow problems with mixing or reaction the program will automatically calculate the fluid specific heat according to the local species concentrations and mixture temperature. For problems with heat transfer but without mixing or reaction the user must specify the constant value of the specific heat. This is done with the command

$$\textbf{SPECIFIC\_HEAT} \left[\begin{array}{l} \text{CONSTANT rval} \\ \text{MIX\_JANNAF} \\ \text{MIX\_POLY5} \end{array} \right]$$

The default setting for specific heat is a constant value of 1000 J/kg·K unless the flow involves mixing or reaction in which case the MIX_JANNAF option is selected.

When the MIX_POLY5 option is selected, the fluid specific heat will be evaluated as a function of the specific heats of the individual species and the local species concentrations. Individual species specific heats are calculated as fifth-order polynomial functions of temperature. Polynomial coefficients must be input for each species with the command

CP_POLY5_COEFS species_name c_0 c_1 c_2 c_3 c_4 c_5

7.5 Conductivity

The CONDUCTIVITY command is used to select the option for fluid conductivity calculation. The command with its various options is as follows:

$$\textbf{CONDUCTIVITY} \left[\begin{array}{l} \text{CONSTANT k} \\ \text{PRANDTL Pr} \\ \text{POLY5} \quad c_0\ c_1\ c_2\ c_3\ c_4\ c_5 \\ \text{MIX\_POLY5} \end{array} \right]$$

The CONSTANT option allows input of a constant value of fluid conductivity, k.

The PRANDTL option allows input of a constant PRANDTL number from which the fluid conductivity will be evaluated based on local viscosity and specific heat.

The POLY5 option allows conductivity to be calculated from a fifth order polynomial in temperature.

$$k = c_o + c_1T + c_2T^2 + c_3T^3 + c_4T^4 + c_5T^5$$

The MIX_POLY5 option allows conductivity to be calculated from individual species conductivities and local species concentrations. Species conductivities are evaluated as fifth-order polynomials in temperature. Polynomial coefficients must be entered for each species with the command

COND_POLY5_COEFS species_name c_0 c_1 c_2 c_3 c_4 c_5

7.6 Solid Properties

For conjugate heat transfer problems, activated by the specification of solid regions in the GEOMETRY section, it is necessary to set the thermal properties of the material. This is done with the SOLID_PROPERTIES command

$$\text{SOLID\_PROPERTIES imat K=}\begin{bmatrix}\text{k}\\ \text{POLY5}\end{bmatrix}\ \text{CP=}\begin{bmatrix}c_p\\ \text{POLY5}\end{bmatrix}\ \text{RHO=rho ABSOR=a EMISS=e}$$

where

imat is the material identifier entered in the SOLID command

k is the thermal conductivity (J/s·m·K)

cp is the specific heat (J/kg·K)

rho is the density (kg/m^3)

a is the absorption coefficient of the material

e is the emissivity to be used at the surfaces of the material.

When the POLY5 option is selected for either conductivity or specific heat or both, then these properties can be evaluated as fifth-order polynomials in temperature. The SOLID_PROPERTIES command must be followed by one or both of the following statements which provide the polynomial coefficients

POLY5_COEFS K c_0 c_1 c_2 c_3 c_4 c_5

POLY5_COEFS CP c_0 c_1 c_2 c_3 c_4 c_5

SECTION 8

MODELS

The MODELS input section is used to select the turbulence model to be employed for turbulent flow simulations, to select the reaction model to be used in simulation of reacting flows and to activate various sub-models associated with the spray model.

8.1 Turbulence Models

There are five turbulence models available in the SCISEAL code. These models are

Eddy Viscosity model,
Baldwin-Lomax model,
Standard k-ε model,
Low Reynolds number model, and
2-Layer k-ε model.

Details of these are provided in the Final Report of the Seals Contract.

The activation of each model is carried out by

$$\textbf{TURBULENCE-MODEL} \begin{bmatrix} \text{EDDY\_VIS value} \\ \text{B\_L} \\ \text{KE} \\ \text{LOW\_RE} \\ \text{2\_LAYER} \end{bmatrix}$$

The k-ε turbulence model is the default model used in SCISEAL and it is recommended that it be used in most cases unless the user is knowledgeable of turbulence modeling and can select a better model. The eddy viscosity model can also be useful as a rough but simple approximation to turbulent flow. It requires specification of a constant eddy viscosity, usually several orders of magnitude larger than the molecular viscosity.

8.2 Turbulence Model Selection for Seals

The typical seal clearances are small, and when selecting turbulence models, care must be taken to ensure that the model being used is valid for the geometry and grid selected. The main parameter is the distance of the first cell center away from the wall boundaries in terms of the nondimensional parameter y^+. Following are the recommended limits on this parameter.

a. **k-ε model with standard wall functions:** This model is applicable for $y^+ > 11.5$. To ensure the accuracy of the wall functions, the first cell center must be sufficiently **away** from the wall. The wall shear is overpredicted if the first cell is too close.

b. **Low Re number k-ε model:** To ensure the accuracy of this model, the first cell center sufficiently **close** to the wall so that it will be inside the laminar sublayer, which corresponds to a y^+ of less than 1.

The wall shear calculations affect the leakage rates, since it affects the pressure drop in the seal. The rotordynamic coefficients are calculated using the circumferential pressure generated by an eccentric rotor. This pressure is also affected by the wall shear calculation. To ensure the correct placement of the cell centers nearest to walls, the **PRINTF** command with **WALL** option can be used to print out the y^+ distances at all wall boundaries, and then these values can be checked for consistency with the turbulence model being used.

In several instances, the y^+ values are not known *a priori* and the flow regime may fall outside of the limits specified by the selected turbulence model, and a second run with appropriate grid change or with a different turbulence model may be needed. To avoid this, the 2-layer model was implemented in the code. The model automatically switches from the low Re k-ε type formulation to a wall function type formulation depending on the near wall y^+ distance. For this reason, this model is recommended whenever there is a likelihood of the near wall distances falling outside the ranges with the first two models.

SECTION 9

BOUNDARY CONDITIONS

The boundary conditions to be applied to the model are specified in the BOUNDARY_CONDITIONS input section. SCISEAL provides the capability to specify inflow, outflow, wall, and symmetry boundaries over surfaces of the calculation domain. Periodic or cyclic boundary conditions can also be activated as can body forces due to gravity and to coordinate system rotation. The commands for boundary condition specification are described in the remainder of this section. Proper specification of boundary conditions is extremely important for both code convergence as well as to get the correct flow problem. Extreme care should be given, therefore, to ensure that the boundary condition specification is proper and correctly reflects the flow problem to be solved.

9.1 Section Format

For multi-domain models the boundary condition input must be provided for each domain in exactly the same order as the domains were specified in the GEOMETRY section. Just as in that section, the DOMAIN command increments the domain number. Input in the BOUNDARY_CONDITIONS section appears as follows

```
BOUNDARY_CONDITIONS

 input for domain 1
 DOMAIN 2
 input for domain 2
 DOMAIN 3

 etc.

 DOMAIN n
 input for domain n
```

END

9.2 Surface Boundary Conditions

Boundary conditions must be applied over each face of the calculation domain and at the surfaces of each blocked region. If the user provides no boundary conditions over any portion of these surfaces, the SCIPRE program will automatically apply a default boundary condition of an adiabatic wall.

Application of surface boundary conditions generally requires the following two steps:

- Specify type of boundary condition and location of region over which it is to be applied.
- Specify boundary values prevailing at the region.

The default boundary condition is extremely useful since it removes the necessity of imposing wall boundaries explicitly. However, as the SCIPRE code doesn't print out any messages concerning the default boundary zones, mistakes in other boundary zone specifications cannot be caught and may lead to erroneous answers. The user, therefore, is urged to ensure that all non-wall boundary conditions are specified correctly.

9.2.1 Location

Boundary conditions are applied over rectangular surfaces in computational space. That is, the region over which a boundary condition is applied can be described by the beginning and ending cell indices.

The general format of the commands to specify the type of boundary condition and to locate the region over which the boundary condition is to be applied is as follows

Keyword i_f i_l j_f j_l k_f k_l (SIDE)

where

Keyword is one of the following mnemonics and indicates the type of boundary condition to be applied

INLET indicates an inflow boundary

TOTAL_P indicates an inflow boundary with specified total pressure

EXIT_P indicates an exit boundary with fixed pressure

EXIT_E indicates an exit boundary with flow conditions extrapolated from interior conditions

EXIT_C indicates an exit boundary with fixed pressure if flow is subsonic but extrapolated pressure if flow is supersonic

WALL indicates a wall boundary

SYMMETRY indicates a symmetry boundary (zero flux)

INTERFACE indicates a domain interface

The other parameters have the following meaning

i_f first i-direction cell index of the region

i_l last i-direction cell index of the region

j_f first j-direction cell index of the region

j_l last j-direction cell index of the region

k_f first k-direction cell index of the region (3D only)

k_l last k-direction cell index of the region (3D only)

The keyword SIDE is optional and is required only under special circumstances. Usually when the boundary region indices are specified, the pre-processor automatically imposes the correct side based on the index information. However, when the starting and ending indices in two or more direction are the same (i.e. a boundary with a single cell in 2-D or a domain with a single cell or a line of cells in 3-D), the automatic side imposition algorithm will fail. In such a case, the SIDE parameter needs to be specified. The allowed keywords are: EAST, WEST, SOUTH, NORTH, and LOW and HIGH (3-D) and correspond to the lower and upper index side in i, j, and k index directions, respectively.

9.2.2 Boundary Values

Constant Boundary Values. Specification of the relevant physical properties, flow conditions, and values for dependent variables at each boundary region is required immediately after the command locating the region for the INLET, TOTAL_P, EXIT_P, and WALL boundary conditions. Specification of the boundary values takes the following form

P = p   U= u   V= v   W = w   T = t   K = k   D = d   L = l   C = c

where

p   is the static pressure (except in the case of the TOTAL_P boundary condition when it is the total pressure)
u   is the x-direction velocity component
v   is the y-direction velocity component
w   is the z-direction velocity component
t   is the static temperature (except in the case of the TOTAL_P boundary condition when it is the total temperature)
k   is the turbulence kinetic energy
d   is the dissipation rate of turbulence kinetic energy
l   is a characteristic length scale (usually an inlet width)
c   is the composition number.

The quantities that must be specified vary depending upon the type of the boundary and the dimensionality and type of model. Table 9-1 identifies the quantities that must be specified for each boundary type. Note that the EXIT_E, SYMMETRY and INTERFACE boundary conditions require no statement of boundary values.

Three quantities related to the turbulence characteristics at the boundary K, D, and L can be specified at inlet boundaries. It is only necessary to provide values for two of these quantities. The program requires boundary values for the turbulence kinetic energy, K, and the dissipation rate, D. If values are provided for both of these, the program will ignore any value provided for length scale, L. It is sometimes more convenient, however, to provide a length scale instead of a value for dissipation rate. In this case, the program will calculate the boundary value of the dissipation rate to be used as a boundary condition.

Table 9-1. Required Boundary Values

BOUNDARY TYPE	VARIABLES
INLET	P, U, V, W, T, K, D, L, C
TOTAL P	Po, To, K, D, L, C, [PFACT]
EXIT P	P [U, V, W, T, K, D, C]
EXIT E	-
EXIT C	P [U, V, W, T, K, D, C]
WALL (adiabatic)	U, V, W , [ROUGH]
WALL (isothermal)	U, V, W, T, [ROUGH]
WALL (heat flux)	U, V, W, Q, [ROUGH]
WALL (external convection)	U, V, W, HC, TINF, [ROUGH]
	U, V, W, TINF
SYMMETRY	-
INTERFACE	-

Notes:		
	C	is composition number
	Q	is heat flux
	HC	is external heat transfer coefficient
	TINF	is external temperature

The conventional entrance pressure with a lost factor boundary condition can be specified using TOTAL_P condition and specifying PFACT as the entrance loss factor.

The default wall surface is assumed to be smooth. When the wall has surface roughness, its effects on wall momentum and heat transfer can be included in turbulent flow simulations by specifying the sand grain roughness height ROUGH. This height should be specified in meters. Such a specification automatically signals the code that the wall is rough.

Following are some special points about specifying boundary condition values that should be considered to maintain consistency:

- Fixed pressure outlet boundaries can also allow inflow. For this reason, it is important to provide realistic values of turbulence quantities, temperature, and composition at these boundaries.

- When using the temperature-dependent or Ideal Gas Law density options, you must provide a realistic temperature at inlet boundaries. Otherwise, the fluid density will be infinite and a solution cannot be obtained.

- When a reference pressure has been provided for the temperature-dependent or Ideal Gas Law density options, then all boundary as well as field solution pressures are relative to that reference pressure.

- You should not specify any boundary conditions adjacent to solid regions as these interface conditions (fluid-solid and solid-solid) are determined automatically by the program.

- An isothermal wall boundary condition can be specified within a solid region either adjacent to a domain boundary or adjacent to a blocked region. These boundary conditions should not span solid regions of different material types, however, nor should they span both a solid region and a fluid region.

- Domain or block interface boundary conditions must match, cell-for-cell, with another interface boundary in the adjacent domain.

Alternative Specification of Boundary Velocities. Boundary velocities are normally specified in a global cartesian reference system, even for BFC models. If it is more convenient to specify the grid-aligned velocity components, this can be done by entering

UI=u_i VJ=v_j WK=w_k

where u_i, v_j and w_k are the grid-aligned or contravariant components.

In the case of two dimensional, axisymmetric geometries when the swirl velocity is solved the boundary value is specified in m/s. The boundary velocity can also be specified in radian/s by entering

WA = w

This is often a more convenient method of setting inlet swirl or wall velocities.

Alternative Wall Heat Transfer Conditions. In addition to applying an isothermal wall temperature or an adiabatic wall boundary condition (by omitting specification of temperature) it is also possible to specify a constant heat flux boundary and an external convection boundary. A constant heat flux boundary is applied by including

Q = q

on the boundary value input line where q is the heat flux in w/m^2. To apply an external convection heat transfer boundary condition the boundary value input line should contain

HC = hc   TINF = tinf

where

hc	is the external heat transfer coefficient
tinf	is the external temperature.

Profile Boundary Values. In addition to providing constant boundary values, it is also possible to specify a one-dimensional profile over certain of the region types, namely

INLET	profiles of U, V, W, K, D and T can be specified
WALL	profiles of T can be specified.

An example of the way in which profile boundary conditions are specified at an inlet is as follows

INLET i_f i_l j_f j_l k_f k_l

$$U = \begin{bmatrix} \text{PROF\_X} \\ \text{PROF\_Y} \\ \text{PROF\_Z} \\ \text{PROF\_R} \end{bmatrix} \quad V = v \quad W = w \quad T = t$$

U n

x_1 x_2 ...x_n

u_1 u_2 ...u_n

where

PROF_X	indicates a variation in the cartesian x-direction
PROF_Y	indicates a variation in the cartesian y-direction
PROF_Z	indicates a variation in the cartesian z-direction
PROF_T	indicates a variation in time

The input lines following the INLET statement provide a set of data points which are used to construct a spline to describe the one dimensional variation of the boundary value. The first line contains the name of the variable, in this case U, followed by the number of data points to be provided. Then the spatial or temporal coordinates are provided followed by the boundary values corresponding to those coordinates. These data points must be provided for each variable for which a profile variation has been indicated. When a profile for more than one variable is input, the data must be in the order U, V, W, K, D, T.

In cylindrical geometries, such as seals, the profiles are often specified along the radius direction, and this does not always coincide with the Cartesian axes. SCIPRE has a provision to enforce such a profile only on the WEST and EAST boundaries (i.e., $i_f = i_l$).

The command for this is

PROF_R	indicates a variation in the radial direction

The pre-processor assumes that the axis of the cylindrical domain coincides with the Cartesian x-axis. At present, the PROF_R command will not work on other boundaries.

9.3 Special Types of Boundary Conditions

There are two special types of boundary conditions for which the input is similar to that previously described but slightly more involved. These are the rotating wall and the whirling wall boundary conditions described below.

9.3.1 Rotating Walls

The location of a rotating wall is specified with the WALL_R command.

WALL_R i_f i_l j_f i_l k_f k_l

where the parameters i_f to k_l are the grid indices describing the extent of the rotating wall region.

The rotational speed and direction are specified by providing the coordinates of a point on the axis of rotation and the rotation vector. This is done on a single input line following the WALL_R command. This input line has the form

CL=(x_o, y_o, z_o) OMEGA=(ω_x, ω_y, ω_z)

where

x_o, y_o, z_o are the coordinates of the point on the axis of rotation

ω_x, ω_y, ω_z define the components of the rtation vector with magnitude in rpm

Note that the WALL_R command is applicable to three-dimensional models only.

Wall rotation in a 2-D flow problem can be specified by using a direction specification of the velocity W or WA as described earlier.

9.3.2 Whirling Walls

A whirling wall is a special condition encountered in seal problems. It is characterized by the solid wall region describing a circle about a specified axis of whirl. The wall also can rotate about its axis at the same time. Such a wall can be specified using

WALL_WHR i_f i_l j_f j_l k_f k_l

where the parameters i_f to k_l are the grid cell indices that describe the extent of the region. The whirl specification and other specifications (wall rotation) is specified on the successive line(s). The whirl of a boundary is specified in terms of the whirl speed and a whirl radius defined in terms of two points CWH and CWHB as follows:

$$\text{WHOMGA} = (\omega_{hx}, \omega_{hy}, \omega_{hz}),\ \text{CWH} = (x_{CO}, y_{CO}, z_{CO}),\ \text{CWHB} = (x_{CBO}, y_{CBO}, z_{CBO})$$

where WHOMGA is the rotation vector for whirl speed. CWH are the coordinates of a point on the whirl axis, and CWHB are the coordinates of a point on the whirl circle. In addition, the other wall parameters including rotating wall specification are also accepted. These can be specified on two successive lines if necessary.

Note that a whirling wall implies that the grids are time-dependent. The resulting problem needs to be solved as either time-accurate with moving grids or as quasi-steady using a transformation to a rotating frame which has a center at CWHB and rotates with an angular speed WHOMGA, specified in rpm.

9.4 Cyclic Boundaries

Cyclic or periodic boundary conditions can be activated at the high and low boundaries in the i-direction or the k-direction with the command

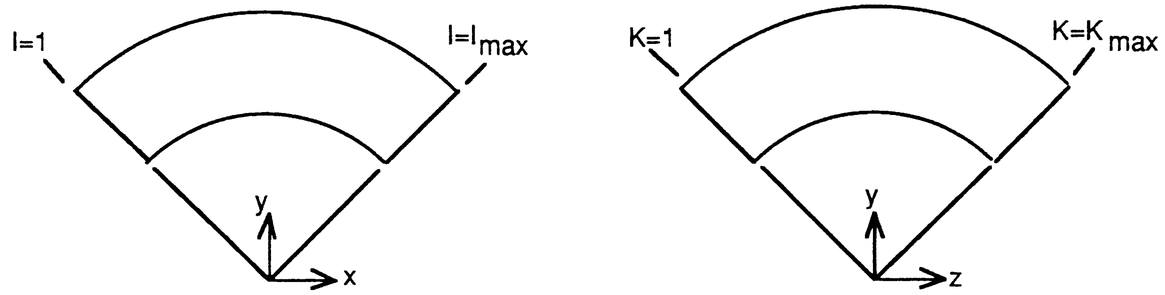
CYCLIC $\begin{bmatrix} \mathrm{I} \\ \mathrm{K} \end{bmatrix}$

Periodic (or cyclic) boundary conditions occur when the calculation domain is confined to just one of a series of repeated regions with identical geometry and boundary conditions. These conditions are particularly likely to occur in cylindrcial geometries with such as seals and turbomachines. SCISEAL allows periodic boundary conditions to be applied in the grid I- and (for three-dimensional models) K-directions. The program imposes the following requirements on cyclic boundary conditions.

1. For I-direction periodicity, the grid at $I=I_{max}$ must be a translation or a rotation of the grid at I=1. The same requirement holds for K-direction periodicity.

2. When I-direction periodicity is applied and the grid at $I=I_{max}$ is a rotation of the grid at I=1, the axis of rotation must be the global Z-axia (see Figure 9-1).

3. When K-direction periodicity is applied and the grid at $K=K_{max}$ is a rotation of the grid at K=1, the axis of rotation must be the global X-axis (see Figure 9-1).



a. Periodic I

b. Periodic K

Figure 9-1. Periodic Boundary Conditions

9.5 Body Forces

Two types of body forces can be specified. Those due to gravity and those due to coordinate system rotation. The way in which to input these is described below.

9.5.1 Gravity

Gravitational forces are specified by providing the x-, y- and z- direction components of the gravity vector with the following commands

GRAV_X rval
GRAV_Y rval
GRAV_Z rval

By default, GRAV_X, GRAV_Y and GRAV_Z are set to 0.0.

9.5.2 Rotation

Coordinate system rotation for three-dimensional models can be activated with the command

ROTATE CL=(x_0, y_0, z_0) OMEGA=(ω_x, ω_y, ω_z)

where the point CL is any point on the axis of rotation and the vector OMEGA defines all three components of the rotation in rpm.

9.6 Momentum Resistances

Momentum resistances are used to simulate the pressure drop due to flow past obstructions within the calculation domain. Resistances can be either concentrated (occurring in a single layer of control cells) or distributed (occurring in some larger subset of the domain). The resistance forces exerted can be either linear or quadratic functions of velocity or both. A concentrated resistance will result in a pressure drop across the layer of cells to which it is applied calculated as

$$\Delta p = \frac{K_l \rho u}{2\beta} + \frac{K_q \rho u^2}{2\beta^2}$$

Note that the linear resistance coefficient, $K_{l,}$ has units of 1/u while the quadratic resistance coefficient K_q is dimensionless. The distributed resistance results in the following relationship between pressure gradient and velocity

$$\frac{\partial p}{\partial x} = \frac{K_l \rho u}{2\beta} + \frac{K_q \rho u^2}{2\beta^2}$$

Momentum resistances can be specified with the following command

$$\textbf{RESISTANCE} \begin{bmatrix} \text{CONC} \\ \text{DIST} \end{bmatrix} \; i_f \; i_l \; j_f \; j_l \; \left[k_f \; k_l \right]$$

$$\text{KXL} = k_{xl} \;\; \text{KYL} = k_{yl} \;\; \text{KZL} = k_{zl} \;\; \text{KXQ} = k_{xq} \;\; \text{KYQ} = k_{yq} \;\; \text{KZQ} = k_{zq} \;\; \text{BETA} = \beta$$

where

CONC indicates a concentrated resistance

DIST indicates a distributed resistance

$i_f \ldots k_l$ are cell indices locating the subset of the domain over which the resistance is to act

k_{xl} is the x-direction linear resistance coefficient

k_{yl} is the y-direction linear resistance coefficient

k_{zl} is the z-direction linear resistance coefficient

k_{xq} is the x-direction quadratic resistance coefficient

k_{yq} is the y-direction quadratic resistance coefficient

k_{zq} is the z-direction quadratic resistance coefficient

β is the porosity of the resistance (flow area/total area)

9.7 Direct Specification Of Source Terms

User specified source terms can be added to the right-hand side of the conservation equations for the selected variables being solved by CFD-ACE. Those source terms can be specified for each dependent variable over a region of the domain using the SETSOR command.

SETSOR type variable i_f i_l j_f j_l k_f k_l coef value

where

type indicates the type of the source term and can be VOLUME, MASS or CELL

variable is the dependent variable name or index

i_f is the index of the first point in the i-direction

i_l is the index of the last point in the i-direction

j_f is the index of the first point in the j-direction

j_l is the index of the last point in the j-direction

k_f is the index of the first point in the k-direction

k_l is the index of the last point in the k-direction

coef is the coefficient

value   is the value.

9.8 Pressure Anchor Points

When incompressible flow solutions are needed and the flow domain contains no boundaries where pressure levels are explicitly stated (Dirichlet condition), the problem can become indeterminant since the actual pressure level is not of concern as long as the gradients are maintained. This situation can arise when all boundaries are of the type INLET, EXTRAP, WALL and SYMMETRY. In such a case a solution can be ensured by specifying an arbitrary point in the flow domain and setting the static pressure to zero at that point. This is done using the following commands

LPDREF idval
LPPREF ipval

LPDREF corresponds to the number of domain that contains the reference point. The default value is 1. LPPREF is the cell index of the reference point in that domain, calculated by "unrolling" the i,j, and k indices of the point in question as:

$$\text{ipval} = (k-1)L^*M + (j-1)^*L + i$$

A default value of 1 is assigned. If more of these have been assigned, the preprocessor will check for the flow conditions and assign the default values if no explicit pressure boundaries are encountered in the flow domain.

9.9 Seal Boundary Conditions

With the grid geometry and orientation as described in Section 5 for cylindrical seals, the location of the various boundaries becomes fixed. The boundaries are divided into several suitable regions, and physical conditions are specified on each of the regions. Cylindrical seals have relatively few boundary regions, and these are shown in Figure 9-2.

The post-processing routine which is used to calculate the integrated quantities such as rotor load and torque assumes the locations and sides of the boundary regions as shown in Figure 12-1. To ensure the accuracy of the integrated quantities, following convention should be followed.

Seal Inlet: at $i_f = i_l = 1$, specify (a) static pressure (EXIT_P), or (b) fluid velocity: constant or profile (INLET) or (c) stagnation pressure with loss factor (TOTAL_P and PFACT).

Seal Exit: at $i_f = i_l = L$, specify (a) static pressure (EXIT_P) or (b) extrapolation (EXIT_E or EXIT-C)

Stator Wall: at $j_f = j_l = M$, stationary wall condition

Rotor Wall: at $j_f = j_l = 1$, wall conditions with wall velocities; rotating wall, or a whirling wall (WALL_R, WALL_WHR)

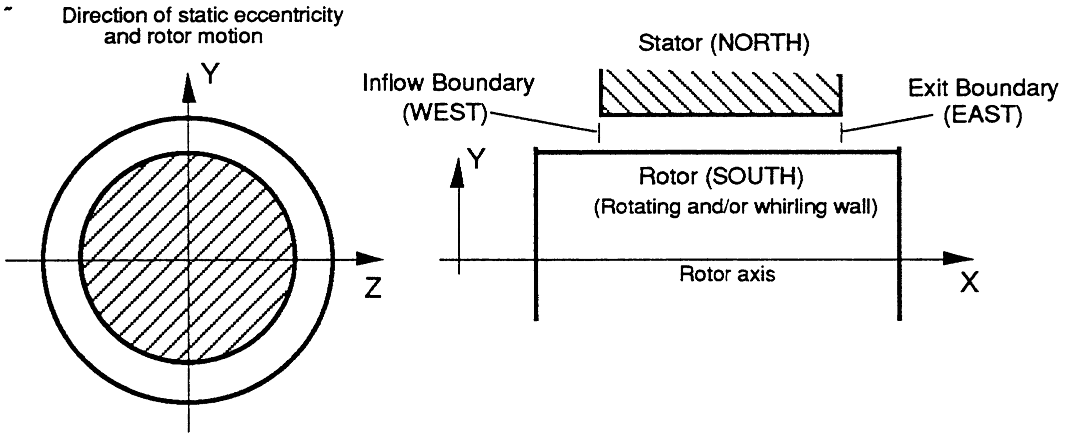


Figure 9-2. Details of a Generic Cylindrical Seal Configuration

SECTION 10

INITIAL CONDITIONS

Initial conditions can be set to be uniform over all of the calculation domain, can be set for each domain of a multi-domain model or can be set over sub-sets or sub-domain of each domain. Initial conditions can also be read from the results file of a previous analysis.

10.1 Uniform Initial Conditions

The initial values of the dependent variables which prevail over the entire calculation domain at the start of a simulation are specified by the values given to the following quantities

U=u V=v W=w P=p T= t K=k D=d L=l C=c

The first statement in the INITIAL_CONDITIONS section must be one that assigns initial values as above. Once that has been done then the user can optionally set other initial values over subsets or zones of the calculation domain. Note that other parameters such as density and enthalpy are calculated by the code based on the specified quantities.

10.2 Multi-Domain Considerations

In multi-domain models, initial conditions can be specified over each domain in a manner identical to the way geometry and boundary conditions are defined for multi-domain models. The input format is as follows

```
INITIAL _CONDITIONS

  Input for domain 1
  DOMAIN 2
  input for domain 2
```

```
DOMAIN 3

etc.

DOMAIN n
input for domain n
END
```

The minimum input for each domain is a single statement specifying the uniform initial conditions to prevail over that domain.

10.3 Initial Conditions Over Sub-Domains

Initial conditions can be specified over sub-sets or sub-domains of any domain with the following two input statements

SUB_DOMAIN i_f i_l j_f j_l k_f k_l
U=u V=v W=w P=p T=t K=k D=d L=l C=c

where the indices i_f to k_l define the extent of the sub-domain.

10.4 Initial Conditions From Previous Solution

As an alternative to specifying initial conditions in the input file, it is possible to use the results of a previous analysis as initial values. This is done with the RESTART FROM command.

RESTART FROM filename

The filename argument is the name of the previously saved solution file.

SECTION 11

SOLUTION CONTROL

This section describes the commands that control the numerical solution process. This entails selection of the solution algorithm, differencing scheme, solution iterations and other sub-iterations, under-relaxation, linear equation solvers and variable limits. Further details of the actions or meanings of the commands in this section are provided in Reference 3.

11.1 Solution Algorithm

Two different solution algorithms are available in this SCISEAL program, the SIMPLEC algorithm and the PISO algorithm. These are selected with the command

$$\textbf{ALGORITHM} \begin{bmatrix} \text{SIMPLEC} \\ \text{PISO} \end{bmatrix}$$

The default algorithm is SIMPLEC. At the current release level the PISO algorithm has not been extensively tested and its use is not recommended.

11.2 Spatial Differencing Schemes

The **S_SCHEME** command determines the type of differencing scheme that should be used to calculate the convective terms in the transport equation for the dependent variables specified. The command has the form

$$\textbf{S\_SCHEME} \begin{bmatrix} \text{UPWIND} \\ \text{SOU} \\ \text{CENTRAL} \\ \text{OSHER} \end{bmatrix} \begin{bmatrix} \text{variable\_list} \\ \text{ALL} \end{bmatrix}$$

where

UPWIND indicates upwind differencing
SOU indicates second order upwind differencing
CENTRAL indicates central differencing
OSHER indicates the Osher-Chakravarthy differencing scheme.

variable list is a list of the dependent variable names for which the specified scheme should be used (see Table 11-1).

The upwind differencing scheme is the default for all variables.

The degree to which the higher-order schemes are incorporated is controlled by a blending factor which is set by the following command

$$\textbf{S\_BLENDING}\ \text{rval} \begin{bmatrix}\text{variable\_list}\\ \text{ALL}\end{bmatrix}$$

The blending factor varies between 0, where the scheme is completely the high-order scheme, and 1 where the scheme is completely upwind. The default blending factor for the higher-order differencing schemes is set to 0.1 for all variables, which results in a differencing scheme which is 10% upwind and 90% higher order.

Table 11-1. Dependent Variables

NAME	DESCRIPTION
U	x-direction velocity component
V	y-direction velocity component
W	z-direction velocity component
PP	pressure correction
K	turbulent kinetic energy
D	dissipation rate of kinetic energy
H	enthalpy
F1	mixture fraction of composition 1
.	
.	
.	
F18	mixture fraction of composition 18

11.3 Temporal Differencing Schemes

Selection of the temporal differencing scheme for transient problems is activated by

$$\textbf{T\_SCHEME} \begin{bmatrix} \text{EULER} \\ \text{CRANK\_NICOLSON} \end{bmatrix}$$

The Euler scheme is the default temporal differencing method and results in first-order temporal accuracy. The Crank-Nicholson scheme is implemented as a general blending or averaging of the old time step fluxes and new time step fluxes, and the blending factor denotes the fraction of the new time step fluxes. The factor is specified as

T_BLENDING value

A value of 0.5 results in the exact Crank_Nicolson scheme but a default value of 0.6 is assigned. The slightly higher value increases the implicitness of the scheme and increases the stability of time-integration algorithm.

11.4 Solution Iterations

The maximum number of iterations that the program performs on the overall solution procedure is set by

```
ITERATIONS ival
```

By default, the program starts the iteration count at 1, even if the initial conditions are taken from a previous solution (perhaps partially converged). If, instead, you wish to have the program start the iteration count at one more than the iteration count from the previous run, you can do so with the command

```
CONTINUE ON
```

The program will also append to the model.RSP file instead of overwriting it making it easier to obtain continuous residual histories through one or more restarts. Note that the program will always perform exactly the number of iterations requested in the ITERATIONS command.

When using one of the rotordynamic coefficient calculation methods, the iterations for the results of each whirl frequency can be set using the command

```
RTD_ITERATIONS    ival
```

If this variable is not specified explicitly, the default value of these iterations are set equal to the ITERATIONS variable described earlier.

11.5 Continuity Iterations

The number of "continuity iterations" that the program performs on the pressure correction equation within each solution iteration is set by

C_ITERATIONS ival

At each continuity iteration the mass residuals are reduced, improving the overall mass balance. By default, this value is set to 1. It is recommended that a larger value should be used for: a) non-orthogonal BFC problems; b) compressible flow problems; and c) fine grids with high aspect ratios. In those flow cases, two to five continuity iterations are typically used.

11.6 Under-Relaxation

The command used to input values of the inertial under-relaxation factor for each dependent variable is

INERTIAL_FACTOR rval variable list

where

rval	is the inertial under-relaxation factor
variable_list	is a list of dependent variable names (see Table 11-1)

Initial under-relaxation is used to impose relaxation during the solution of momentum equations and the equations for other variables. An inertial under-relaxation value of 0.0 signifies no under-relaxation. The default value is 0.2 for all momentum equations and 0.4 for other variables. Inertial under-relaxation for pressure comes from the momentum equations, and hence, cannot be specified. Higher under-relaxation may be needed in the initial part of a solution procedure due to the effect of the initial conditions.

In addition, each of the primary and auxiliary variables can be under-relaxed during the iterative procedure using direct under-relaxation. This number is a fraction and signifies the proportion of the new iteration value and last iteration value of the variable that is used to calculate the updated new iteration value.

The command used to input linear under-relaxation factors for the auxiliary variables is

RELAX rval variable list

where

rval is the linear under-relaxation factor
variable_list is a list of the auxiliary variable names (see Table 11-2)

The value of linear relaxation factors vary between 0. (heavy linear relaxation, solution frozen for specified variable) and 1.0 (no linear relaxation). Default values are 1.0. Typically, the pressure, viscosity in turbulent flows and temperature may need heavier under-relaxation during the initial part of an iterative solution when the initial conditions are drastically different from the steady state value. This is true especially for the initial k and ε fields which can result in unrealistic turbulent viscosity and cause solution divergence. This can be remedied by using direct relaxation on the viscosity.

Table 11-2. Auxiliary Variables

NAME	DESCRIPTION
RHO	Density
P	Pressure
T	Temperature
VIS	Viscosity

11.7 Solver Selection And Iterations

The command to specify the type of linear equation solver to be used for a specific dependent variable is

```
SOLVER ⎡WHOLE_I ⎤ variable¯list
       ⎢WHOLE_J ⎥
       ⎢WHOLE_K ⎥
       ⎣CG      ⎦
```

The WHOLE solvers are based on a strongly implicit iterative solution method, and the indices, I,J, or K represent the index directions in which line implicitness is used. The default direction is I. The implicitness index should be selected based on the predominant flow direction. Although any solver can be used, with appropriate implicitness the solver performance can be enhanced. The conjugate gradient solver is intended primarily for energy equation solutions on conjugate heat transfer problems. It is useful in solving the stiff linear equation that arises from the energy balance; it may also be needed for pressure correction equation in certain multi-domain problems with disparate grid aspect ratios.

The number of iterations that the linear equation solver performs on a specified dependent variable is controlled by the command

```
S¯ITERATIONS  ival ⎡variable_list⎤
                   ⎣     ALL     ⎦
```

where

ival   is the number of solver iterations

By default, the number of solver iterations is set to 4 for all dependent variables except for the pressure correction where it is set to 10. It is recommended that it be increased only when convergence problems occur. The solver performance can be checked using the INFO=1 option in the OUTPUT section. This option should be used with caution, since the default output file that stores this information can become very large in size.

11.8 Variable Limits

The allowable minimum and maximum values a variable can assume are set by

MINVAL rval variable list
MAXVAL rval variable list

The linear equations are considered solved when the maximum difference value drops at least six orders of magnitude for the WHOLE_I,J,K solvers. Usually all variables except pressure correction will not require more than 10 solver steps. The pressure correction equation, however, may need higher number of iterations; up to 20 or 30. If the continuity equation convergence is slow, increased number of solver sweeps or continuity iterations may help.

SECTION 12

OUTPUT

This section contains the commands that control the output of the preprocessor and the analysis modules. This printout is written in the default output unit and can be written directly to a disk.

12. 1 Field Printout

The command which determines the variables for which initial and final field printouts are to be provided is

PRINT variable_list

All of the dependent and auxiliary variable names (see Tables 11-1 and 11-2) can be included in the variable list. Additionally, when mixing or reacting flow simulations are performed, any of the chemical species listed in Chapter 7, (Compositions) can be printed.

The command which determines the variables for which only final field printout is to be provided is

PRINTF variable_list

The variable names are the same as those of the PRINT command. One additional keyword, WALLS, can be included in the variable list. This will cause the analysis program to print quantities such as y^+ and u^+ near the wall for turbulent flows.

The numbers of columns and rows that are to be skipped from field printout can be set by

IPSKIP ival
JPSKIP ival

```
KPSKIP  ival
```

By default, IPSKIP=JPSKIP=KPSKIP=1, which means that the whole field is to be printed.

The time step frequency of field printout for transient simulations is set by

```
TSKIP      ival
```

The default value of TSKIP is 1 which provides for field printout at every time step.

12.2 Monitor Output

Monitor output, which records the variation of dependent or auxiliary variables with iteration (steady-state) or time (transient) can be requested with the command

```
MONITOR  domain  i  j  [k]  variable_list
```

where

domain — is the number of the cell at which values are to be monitored
i — is the i cell index
j — is the j cell index
k — is the k cell index (3D problems only)
variable_list — is a list of up to five dependent and/or auxiliary variables

The output for each monitor command (up to 20) will be written to a file called model.ii.MON where ii is the number of the MONITOR command.

12.3 Residual Output

The dependent variables for which residuals will be written to the residual listing and plotting files (model.RSL and model.RSP) can be set with the command

```
RES_PRINT  variable_list
```

By default, the program activates residual printout for u, v, w, PP, k, d, H and F1. The RES_PRINT command is usually used to request output of residuals for mixture fractions beyond F1.

12.4 Solution Files

12.4.1 Steady State

The frequency of saving the restart file (and graphics files) can be set with the command

```
RESTART_SAVE iter
```

The program will save a copy of the current solution every iter iterations. The default value of iter is 10000 so that the restart file is usually saved only at the end of execution.

12.4.2 Transient

In transient simulations, the time step frequency of saving the solution files can be set with the command

```
TIME_SAVE   istep
```

A unique name for the files saved at each time step can be requested with the command

```
UNIQUE_NAME    ON
```

12.5 Graphics Files

Two commands are available to allow control of the PLOT3D files generated by the analysis code and by the preprocessor. Generation of the grid, vector and scalar files by the analysis module is controlled by the PLOT3D command

PLOT3D $\begin{bmatrix}\text{ON}\\\text{OFF}\end{bmatrix}$ [UNFORMATTED]

The default setting is ON so that the files will be created. The optional trailing parameter UNFORMATTED can be used to create unformatted rather than formatted data files. Unformatted files are smaller but may not be transportable across all computer systems.

When the PLOT3D graphics output is activated, the program will create a grid file (model.PFG or model.PUG), a vector file (model.PFV or model.POV) and a single scalar file (model.PFS or model.PUS). The contents of the scalar file can be set with the SCALAR_FILE command and a second scalar file can be requested with this command. The command format is

SCALAR_FILE i variable_list

where

i is the number of scalar files, either 1 or 2

variable_list is a list of up to five variable names to be included in the scalar file

Variable names can be any dependent or auxiliary variable or any listed in Table 12-1.

Table 12-1. Additional Keywords for SCALAR_FILE Command

NAME	DESCRIPTION
CP	Specific heat
VIS_L	Dynamic viscosity (laminar)
TOT_P	Total pressure
UPLUS	U+ (at wall boundaries only)
YPLUS	Y+ (at wall boundaries only)
H_FLUX	Surface heat flux (at wall boundaries with a thermal boundary condition)

The preprocessor will create a PLOT3D 'wall file' for three-dimensional models with blockages. The user can control the creation of this file with the command

$$\textbf{WALL\_FILE} \begin{bmatrix} \text{ON} \\ \text{OFF} \end{bmatrix}$$

The default is ON so that the file model.WAL will be created. Note that the walls that will be written to the file consist of all default walls that are interior to the calculation domain plus all user-declared walls.

12.6 Grid File Format

The format of the grid file transferred to the analysis code can be set with the command

$$\textbf{GRID\_FILE} \begin{bmatrix} \text{FORMATTED} \\ \text{UNFORMATTED} \end{bmatrix}$$

If the command is omitted the default grid file format is unformatted.

12.7 Seal Output

There are several integrated quantities that are useful in seal flows. These are: (1) forces on the rotor, (2) frictional torque on the rotor, (3) frictional power loss, (4) leakage mass flow, and (5) rotor attitude angle for seals/bearings with eccentricity. To invoke any of these, the following logical constant sould be switched ON. These parameters will be calculated at the end of the steady-state run, and printed out in the default text output file (stload.out).

Rotor load:

RLOAD ON

Rotor torque:

RTORK ON

Rotor power:

RPOWR ON

Attitude angle:

ATTANG ON

The mass fluxes (leakage flow rate) through all inflow and outflow boundaries are printed at the end of the run unconditionally. As the rotor power calculation requires torque calculation. If **RTORK** is set to T, the pre-processor will set RTORK to T as well. Further, to calculate the rotor power, the rotor angular speed is needed. This is extracted from the rotating wall boundary. This requires that the rotor wall speed should be specified through the **WALL_R** command described in a previous chapter. Both the torque and power calculations are done assuming that the axis of rotation of the seal is the Cartesian x axis. This rotation is consistent with that shown in Figure 5-8.

When seal specific quantities are requested, SCISEAL opens separate output files with specific names directly on the disk and stores the requested information. The files that are opened automatically and the data that is stored in these files are:

stload.out : stores mass flow rates through open boundaries, rotor pressure force, torque and power, and rotor attitude angle.

kctrn.out : rotordynamic coefficients for cylindrical seals, calculated using the whirling rotor method

kcpert.out : rotordynamic coefficients for cylindrical seals; calculated using the perturbation N-S method

To print out the mass flow rates through various open boundaries the command

```
M_FLOW ON
```

should be included in the output section. The output identifies every open boundary by its zone, region number and type, and prints the massflow in kg/s. For a 2D planar problem, a unit depth in the third direction is assumed. For a 2D axisymmetric problem, the depth is taken as 1 radian. This implies that to get the mass flow rates in the complete 360° problem, the output flow rates should be multiplied by a factor of 2π.

SECTION 13

GUIDELINES

This chapter is intended to provide a set of recommendations for use of the SCISEAL program. A few important points about the problem setup/preparation task are given and then guidance about the use of solution methods and solution control parameters are offered.

13.1 Problem Definition

Outlined in this manual are all the commands that are needed to set up any seal problem for execution on the SCISEAL code. However, there are a few general points that should be kept in mind during the generation of grids and defining the flow problem. These are discussed below.

13.1.1 Grids

There are several points to keep in mind when performing the grid generation for a new model. They are:

- SCISEAL requirements for two-dimensional grids and for grids when cyclic boundary conditions are to be used.
- The need to avoid coincident points.
- The need to avoid flow passages of only a single cell width.
- The impact of grid non-orthogonality on convergence.

These topics are discussed further below.

In general, SCISEAL can accept any structured grid generated by an external program. There are some points to be kept in mind which were discussed in Section 5 are are repeated here for emphasis and case of reference. The first restriction is that two-dimensional grids must be constructed in the x-y plane; all z-coordinates must be zero. Secondly, when using cyclic-I boundary conditions in 3-D grids and the plane at $I=I_{max}$ is a rotation of the plane at I=1, the axis of rotation must be the Cartesian z-axis. Similarly, when using cyclic-K boundary conditions and the plane at $K=K_{max}$ is a rotation of the plane at K=1, the axis of rotation must be the Cartesian x-axis. Finally, any grid must have a locally right-handed coordinate system. If i j, and k are unit vectors constructed at the vertices of the grid and pointing in the grid I-, J-, and K-directions, then for two-dimensional grids we must have

$$i \times j > 0$$

and for three-dimensional grids

$$(i \times j) \cdot k > 0$$

Another point to be aware of is that the program may have difficulty dealing with coincident grid points such as occur at the center line of a cylindrical-polar grid. Grids requiring coincident points should be created with the points spread over a small but finite distance. In the cylindrical-polar example this could be provided for by specifying a very small inner radius. An inner radius as small as 10^{-6} m, usually too small to be seen, will suffice.

When modeling problems with narrow flow passages it is often difficult to pack many cells into each passage. Nonetheless, you should never generate a grid, or define blockages on a grid, such that there is only a single cell across the width of a flow passage. At least two cells are required between boundaries and/or blockages so that the program can extrapolate pressures to the blockage surface or to the boundary. If the k-ε turbulence model or one of its variants is used, you should not have less than three cells across the width of a flow passage. Otherwise, the wall functions will fix the values of k and ε to unrealistic values. Note that three cells is too few for a reasonable solution; at least ten cells would be much better.

One of the most important points to keep in mind during grid generation is the beneficial effect of grid orthogonality on convergence of the solution process. While solutions have been obtained with grids having included angles as small as 2°, it is best to try to create grids with included angles no smaller than 25°. Obtaining converged solutions becomes increasingly more difficult as the angle decreases.

13.1.2 Boundary Conditions

In the standard format, boundary conditions are given as surfaces or grid planes, i.e., as rectangular regions of grid vertices with a constant I-, J-, or K-index. When this information is passed on to the preprocessor, it is converted to a rectangular region of cells as well as a specification of the surface (or side) of the cells through which the boundary condition acts. The GUI, then, has to determine the layer of cells the boundary condition is to be applied to (for a surface interior to the domain, there are two candidates). This is straightforward when the boundary condition is located at the domain boundary or on the surface of a blockage. Sometimes, however, more than one of the I, J, or K indices are constants, e.g., for a line boundary in 3-D, in which case, the side of the boundary face must also be specified, e.g., WEST or EAST.

A second important point to note (also discussed in Chapter 4) is that it is important to specify 'reasonable' boundary values of all flow variables at fixed pressure exit boundaries. This is due to the fact that inflow can occur at these boundaries either temporarily during the course of the solution process or at the final solution. When this happens, the boundary values will be transported into the calculation domain. If the boundary values provided are not reasonable then their effect can be to corrupt the final solution or even to cause the solution process to diverge.

13.1.3 Initial Conditions

When steady state simulations are performed, the initial conditions are merely an initial guess and do not influence the final solution. They can, however, influence the convergence path to the final solution and it is still important to provide reasonable initial values for the dependent variables. This is particularly true for pressure, the values of k and ε when that turbulence model is used, and for density when a variable density is allowed. Initial pressures should usually be set to the exit pressure when fixed mass flow rate inlets are used but should be set to the inlet pressure when total pressure inlet boundaries are used. The values of k and ε should be set such that the turbulent viscosity is greater than or equal to the laminar viscosity. A reasonable value of density is provided by specifying reasonable values for temperature and pressure.

13.2 Solution

Because of the non-linear nature of the governing equations and the wide variety of flow problems that can be posed, the convergence of the iterative solution process cannot be guaranteed. There are several parameters that can be adjusted to assist the convergence of the solution and these are discussed in this section.

13.2.1 Differencing Schemes

The first order upwind differencing scheme, while the least accurate of the schemes available, is the most robust. If difficulties are encountered obtaining a solution with the higher-order schemes it is usually best to obtain a solution with the upwind scheme and use that solution as the initial guess for a subsequent solution using the higher order scheme. It is also recommended that you use the second order upwind scheme for incompressible flows and the central differencing scheme or the Osher-Chakravarthy scheme for compressible flows when higher order spatial accuracy is needed.

13.2.2 Solution and Continuity Iterations

The number of solution iterations required for convergence is highly problem dependent and can vary from as few as 50 to many thousands. The residual reduction obtained is the true indicator of solution convergence. With the standard double-precision installation of the analysis program, four orders of magnitude reduction in the residuals is usually sufficient to consider the flow converged.

The default setting of one continuity iteration is usually satisfactory for most problems particularly those involving incompressible flows. For problems involving compressible flow or with highly skewed BFC grids it may be necessary to set a higher number of continuity iterations, two usually being sufficient. Be aware that a large portion of the computer time is required for the pressure correction solution and any increase in the continuity iterations will usually result in a noticeable increase in time required for solution.

13.2.3 Solver Iterations

For single domain problems with coarse to moderate grids, the whole-field solvers (WHOLE-I, WHOLE-J, and WHOLE-K) usually obtain sufficiently converged solutions of the linear equations within five iterations for the pressure correction equation and two to three iterations for other variables. As grid size increases the number of iterations required may also increase to 15 to 20 iterations for pressure correction and up to fifteen iterations for other variables. For multi-domain models the number of iterations should be set to at least 15 for the pressure correction equation and at least eight for all other variables.

The conjugate-gradient solver generally requires more iterations for convergence than do the whole-field solvers. Unlike those solvers, though, the conjugate-gradient solver will automatically terminate when convergence is obtained. For that reason, it is usually best to just request a large number of iterations, say 500, for the conjugate gradient solver. Because the conjugate gradient solver does obtain a complete solution to the equations at each solution iteration, it does consume more computer time than the whole-field solvers. It should generally be used only for energy equation in conjugate heat transfer problems, and continuity equation when cell aspect ratios differ a great deal. In most cases, all other equations can be treated with the WHOLE field solvers.

13.2.4 Under-Relaxation

The proper setting of under-relaxation factors for dependent and auxiliary variables is also highly problem dependent but some general suggestions can be provided. For most single fluid or mixing problems the setting of inertial under-relaxation factors in the range of 0.2 to 0.5 for velocities will be sufficient. When the k-ε turbulence model is used it may be necessary to set under-relaxation factors for k and ε slightly higher than for the velocities, say 0.6 when the velocity factors are set to 0.4. It is also typical to under-relax the turbulent viscosity with a linear relaxation factor of about 0.3 and also to place a maximum value limit on the turbulent viscosity (1.0 being a typical limit). Setting of linear under-relaxation for pressure is usually not required unless the grid is highly skewed or the flow is compressible in which case you may need to set the relaxation factor as low as 0.2. Compressible flows may also require under-relaxation of temperature and density to the same degree as pressure. Turbulent flows may also require linear under-relaxation of the viscosity. Reacting flow problems can generally be controlled by setting inertial factors on dependent variables in the ranges described above and linear under-relaxation factors on auxiliary variables to be set as low as 0.2 for pressure and density and 0.1 for temperature.

SECTION 14

REFERENCES

1. Akima, Hiroshi, "A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures", Journal of the Association for Computing Machinery, Vol. 17, No. 4, October 1970, pp 589-602.

APPENDIX A

GRID FILE FORMAT

APPENDIX A

GRID FILE FORMAT

A.1 Formatted Grid File

The data is written in PLOT3D multigrid format as defined below.

RECORD 1

NDOM

FORMAT : FREE

NDOM=integer variable set to the number of domains

RECORD 2

L_1 M_1 N_1 L_2 M_2 N_2 ... L_{NDOM} M_{NDOM} N_{NDOM}

FORMAT : FREE

L_i=number of grid points in the i-direction in domain i;
M_i=number of grid points in the j-direction in domain i;
N_i=number of grid points in the k-direction in domain i;

RECORD 3 TO 2+NDOM

X_1, ... ,X_{LMN}, Y_1, ... ,Y_{LMN}, Z_1, ... ,Z_{LMN}, $IBLANK_1$, ... ,$IBLANK_{LMN}$

FORMAT : FREE

X, Y, AND Z are the grid coordinate data for domain 1. The order of the coordinates is that given by varying the i-index most rapidly, the j-index next most rapidly and the k-index least rapidly. This grid coordinate data is repeated for rest of the domains, one record at a time.

A.2 Unformatted Grid File

Unformatted grid files are written in the same record-wise fashion as are formatted files, but the data are written in FORTRAN binary form.

A.3 Format Differences for 2-D and 3-D Grids

The SCISEAL code is set up to treat 2-D as well as 3-D problems. The grid I/O, however, was kept in the 3-D format for a unified grid format. Due to this reason, the 2-D grids need to be 'converted' to look like 3-D grids. The steps involved are:

a. Set all N_i indices to 1, i.e., use only a single plane in the 'k' direction;
b. Generate all the grids in x-y plane only, and set the z coordinate for all grid points equal to zero. This implies a planar grid in the Cartesian x-y plane; and
c. Write out this modified grid in the standard 3-D format, outlined in this Appendix.

When the 2-D switch is used in the GEOMETRY section of the input file, the N_i indices and z-coordinates of the modified grid will be ignored by the SCISEAL code, and the problem is treated as a true 2-D problem.

When a 2-D grid is output from SCISEAL for plotting (PLOT3D form), a similar conversion is used internally to dump out a 3D grid with a single plane in the k-direction.

APPENDIX B

THERMOCHEMICAL DATA FILE FORMAT

APPENDIX B

THERMOCHEMICAL DATA FILE FORMAT

The user can specify his own data file containing thermochemical data for chemical species not included in the code. The data file must contain 5 records for each specie. The contents of each record are as follows

Record	Data	Format
1	Name, chemical composition, molecular weight.	A18, 6X, 4(A2, I3), 21X, E10.0
2	Coefficients a_1 - a_5 for upper temperature limit.	5E15.0
3	Coefficients a_6 and a_7 for upper temperature limit. Coefficients a_1 - a_3 for lower temperature limit.	5E15.0
4	Coefficients a_4 - a_7 for lower temperature limit.	4E15.0
5	Two coefficients for viscosity calculation.	2E15.0

APPENDIX C

SCISEAL FILES

APPENDIX C

SCISEAL FILES

Name	Description
model.in	Preprocessor input file
model.AUR	Solution results file (Restart file)
model.AUG	Unformatted grid file
model.INT	Domain interface data file
model.RSL	Residual listing file
model.RSP	Residual plotting file
model.out	Analysis program output file
model.VAR	Variable name file
model.WAL	Wall definition file
model.PFG	Formatted grid file
model.PFV	Formatted vector file
model.PFS	Formatted scalar file
model.PUG	Unformatted grid file
model.PUV	Unformatted vector file
model.PUS	Unformatted scalar file
model.nn.MON	Monitor point file

APPENDIX D

UNITS

APPENDIX D

UNITS

SCISEAL employs SI units. Relevant units for primary and secondary quantities are provided below.

Primary Units

Length	meters, m
Mass	kilograms, kg
Time	seconds, s
Temperature	Kelvin, K

Secondary Quantities

Force	Newton, N	$1\ N = 1\ kg \cdot m/s^2$
Energy	Joule, J	$1\ J = 1\ N \cdot m$
Power	Watt, W	$1\ W = 1\ J/S$

Fluid Flow and Heat Transfer Quantities

Density	kg/m^3	
Dynamic viscosity	$kg/(m \cdot s)$	
Kinematic Viscosity	m^2/sec	
Specific Heat	$J/(kg \cdot K)$	
Conductivity	$J/(s \cdot m \cdot K)$	
Latent Heat	J/kg	
Pressure	Pascal, Pa	$1\ Pa = 10^5\ N/m^2$
Velocity	m/s	
Enthalpy	J/kg	
Kinetic Energy	$m^2/s^2 = J/kg$	
Dissipation rate of kinetic energy	m^2/s^3	
Heat Flux	W/m^2	

APPENDIX E

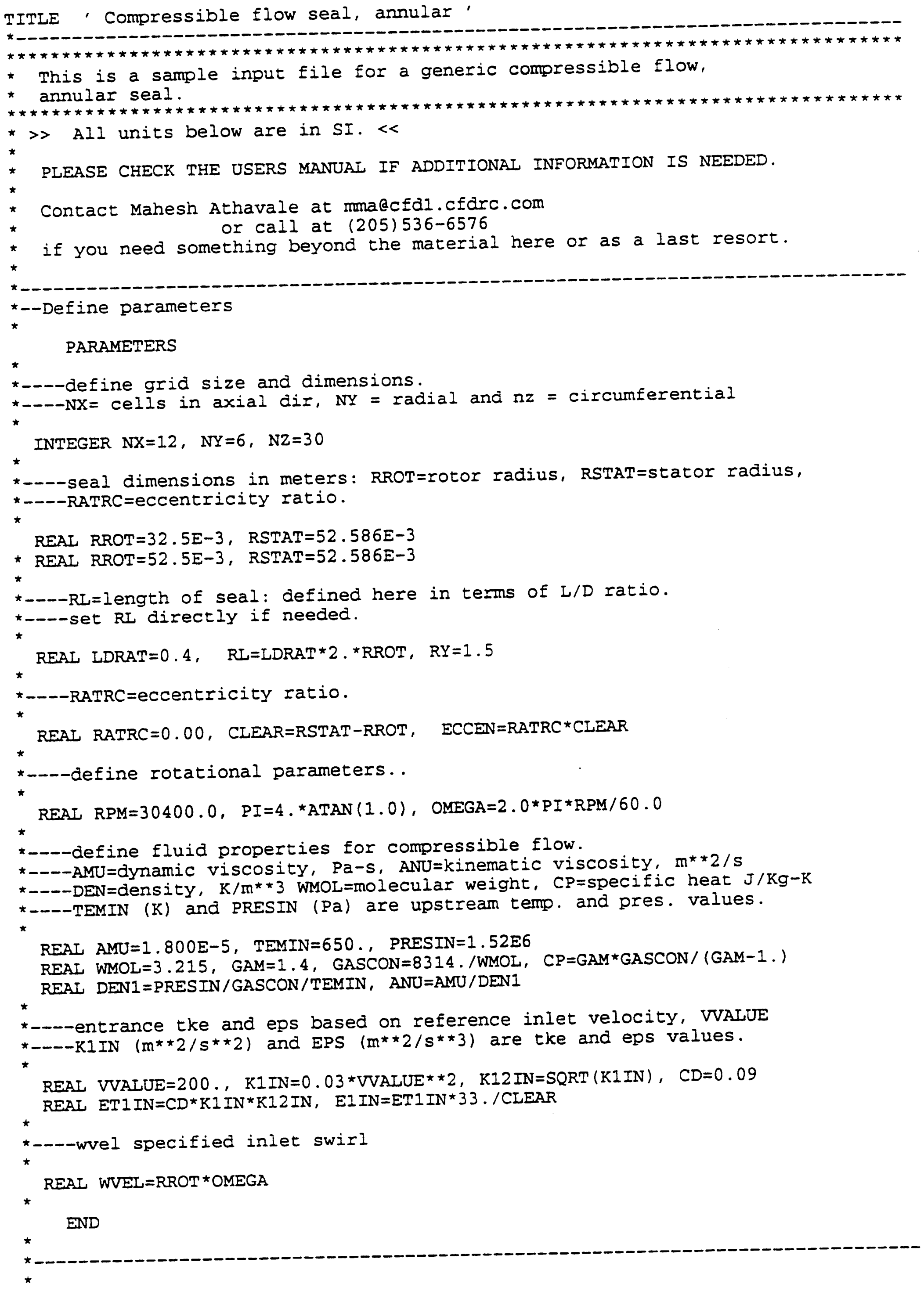
SAMPLE INPUT FILES

Sample Input File for an Incompressible Flow Annular Seal

```
TITLE  ' Incompressible flow seal, annular '
*-----------------------------------------------------------------------------
*******************************************************************************
*  This is a sample input file for a generic incompressible flow,
*  annular seal.  Static eccentricity can be zero.
*******************************************************************************
* >>  All units below are in SI. <<
*
*  PLEASE CHECK THE USERS MANUAL IF ADDITIONAL INFORMATION IS NEEDED.
*
*  Contact Mahesh Athavale at mma@cfdl.cfdrc.com
*                   or call at (205)536-6576
*  if you need something beyond the material here or as a last resort.
*
*-----------------------------------------------------------------------------
*--Define parameters
*
    PARAMETERS
*
*----define grid size and dimensions.
*----NX= cells in axial dir, NY = radial and nz = circumferential
*
  INTEGER NX=10, NY=5, NZ=30
*
*----seal dimensions in meters: RROT=rotor radius, RSTAT=stator radius,
*----RATRC=eccentricity ratio, RL=seal length
*
  REAL RROT=50.0E-3, RSTAT=80.36E-3,  RL=40.0E-3
* REAL RROT=80.0E-3, RSTAT=80.36E-3,  RL=40.0E-3
  REAL RATRC=0.00, CLEAR=RSTAT-RROT,  ECCEN=RATRC*CLEAR
  REAL RY=1.5
*
*----define rotational parameters..
*
  REAL RPM=4000.0, PI=4.*ATAN(1.0), OMEGA=2.0*PI*RPM/60.0
*
*
*----define fluid properties for compressible flow.
*----AMU=dynamic viscosity, Pa-s, ANU=kinematic viscosity, m**2/s
*----DEN=density, K/m**3
*
  REAL AMU=1.000E-3, DEN=1000.0, ANU=AMU/DEN
*
*----entrance tke and eps based on reference inlet velocity, VVALUE
*----K1IN (m**2/s**2) and EPS (m**2/s**3) are tke and eps values.
*----you could assign these values directly to K1IN and E1IN if known.
*
  REAL VVALUE=12.5, K1IN=0.03*VVALUE**2, K12IN=SQRT(K1IN), CD=0.09
  REAL ET1IN=CD*K1IN*K12IN, E1IN=ET1IN*2./CLEAR
*
*----wvel specified inlet swirl
*
  REAL WVEL=RROT*OMEGA
*
  print wvel
    END
*
*-----------------------------------------------------------------------------
*
    GEOMETRY
*
*----define grid type
*
  GRID 3D BFC
  L  NX ;  M  NY;  N  NZ
*
```

```
*
  GCSEAL 1 RL RROT RSTAT RY RATRC 0.0 0.0
*
    END
*
*---------------------------------------------------------------------------
*
    PROBLEM_TYPE
*
*----define variables to be solved:
*
  SOLVE FLOW TURBULENCE
*
*---- activate following if rotordynamics is needed
*----*** PLEASE NOTE>>>> IMPORTANT
*----for a centered seal, i.e. with skew-symmetric coefficient matrix,
*----the circular whirl orbit method is the best.  use the following
*----command:
*
* ROTDYN  TRANSF  RPM  CLEAR 0.1
*
*----use the perturbation method for nominally eccentric seals.
*
    END
*
*---------------------------------------------------------------------------
*
PROPERTIES
*
*----constant viscosity and  density
*
  VISCOSITY CONSTANT ANU
  DENSITY CONSTANT DEN
*
END
*
*---------------------------------------------------------------------------
*
MODELS
*
*----default model for k-e used : Std. k-e with wall functions.
*
  TURBULENCE_MODEL KE
* TURBULENCE_MODEL 2_LAYER
*
END
*
*---------------------------------------------------------------------------
*
BOUNDARY_CONDITIONS
*----we choose to define each boundary region explicitly.
*----WARNING: if there are any boundary regions left unspecified, the
*---- preprocessor will impose a stationary, adiabatic wall on these regions,
*---- and will *not* inform the users.
*
*----stator wall
  WALL     1    L    M    M    1    N
   U=0.0  V=0.0  W=0.0
*----rotor wall, specify rotation
  WALL_R   1    L    1    1    1    N
   CL=(0.0,0.0,0.0) OMEGA=(RPM,0.0,0.0)
*----inlet total pressure condition + loss factor
  TOTAL_P  1    1    1    M    1    N
   W=0. U=16.5 V=0. P=1.0E6 K=K1IN D=E1IN ROUGH=RGHH
*----downstream static pressure condition
  EXIT_P   L    L    1    M    1    N
  U=16.5 V=0.0 W=0.0 P=0.0 K=K1IN  D=E1IN
*
*----specify cyclic conditions on the LOW and HIGH boundaries
  CYCLIC K
*
END
*
*-----------------------------------------------------------------------------
*
INITIAL_CONDITIONS
*
  W=0.0  U=16.50 V=0.0 P=1.2E5 K=0.8167 D=2.2140
*
END
*
*-----------------------------------------------------------------------------
*
SOLUTION_CONTROL
  ALGORITHM SIMPLEC
*----spatial scheme for variables. Use CENTRAL with perturbation method.
  S_SCHEME CENTRAL  U V W
*----blending function for central differencing
  S_BLENDING 0.1 U V W
*----Inertial relaxation (DTFALS)
  INERTIAL_FACTOR 0.2 U V W
  INERTIAL_FACTOR 0.4 K D
*----direct relaxation on pressure and turbulent viscosity
  RELAX 0.6 VIS P
*----select whole-field linear equation solver.
  SOLVER WHOLE_Z U V W PP K D
*----number of iteration for the steady-state solution
  ITERATIONS 60
*----number of iterations for rotordynamics at each whirl frequency
  RTD_ITERATIONS  40
*----number of solver sweeps (iterations) for iterative linear eq. solver.
  S_ITERATIONS 8 U V W K D
  S_ITERATIONS 20 PP
*----number of continuity iterations. set 1 for incompressible flows.
  C_ITERATIONS 1
*----set maximum and minimum values for selected variables for safety.
  MAXVAL 10.0 VIS
*
END
*
*-----------------------------------------------------------------------------
OUTPUT
*----print massflows through open areas
 M_FLOW ON
*----calculate and print rotor torque, power dissipation and pressure load
 RTORK ON
 RLOAD ON
 RPOWR ON
*----print wall info
*----if field print of additional variables is needed, add to this list.
 PRINTF WALLS
END
```

Sample Input File for a Compressible Flow Annular Seal

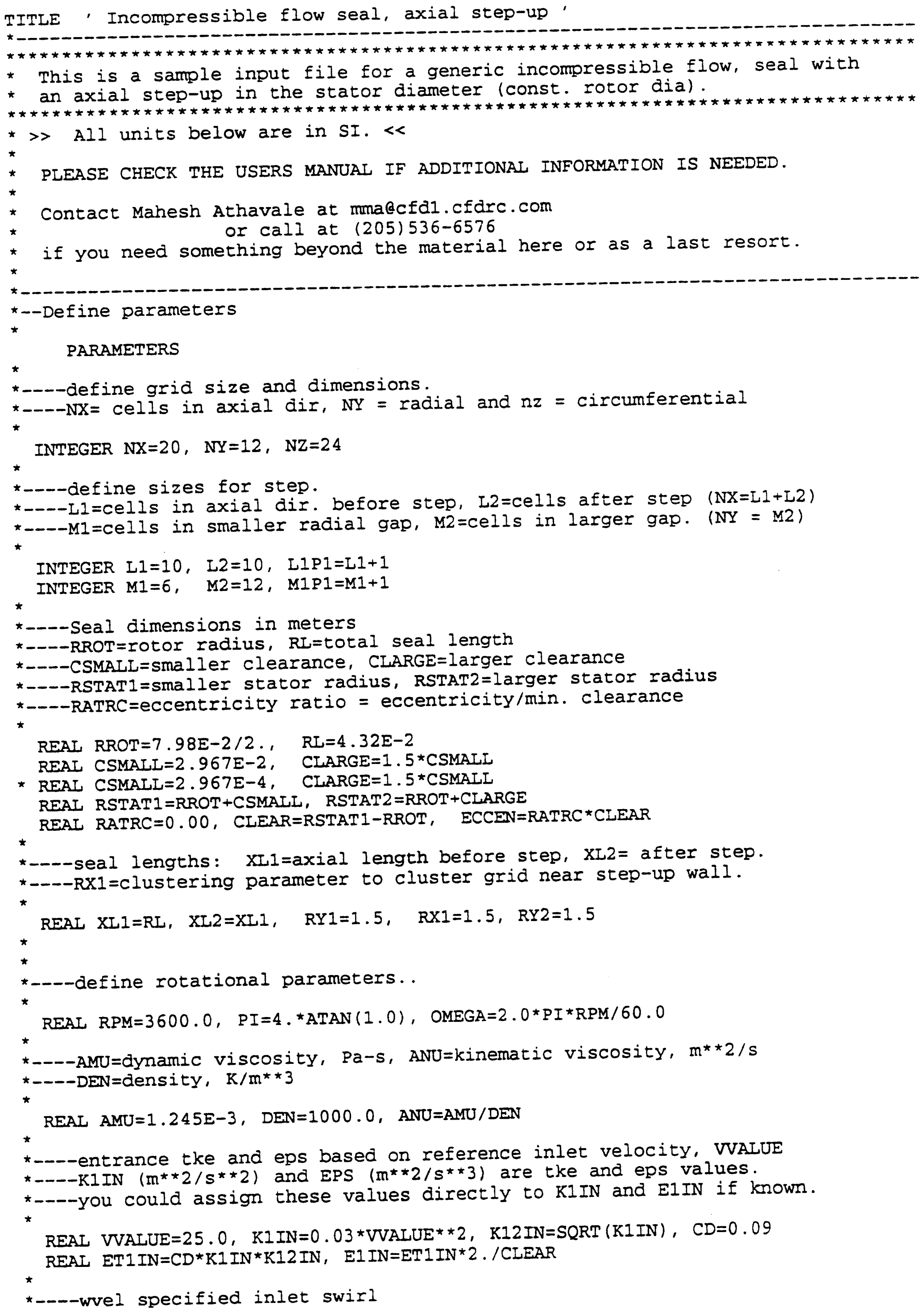


```
TITLE  ' Compressible flow seal, annular '
*------------------------------------------------------------------------------
*******************************************************************************
*  This is a sample input file for a generic compressible flow,
*  annular seal.
*******************************************************************************
* >>  All units below are in SI. <<
*
*  PLEASE CHECK THE USERS MANUAL IF ADDITIONAL INFORMATION IS NEEDED.
*
*  Contact Mahesh Athavale at mma@cfdl.cfdrc.com
*                  or call at (205)536-6576
*  if you need something beyond the material here or as a last resort.
*
*------------------------------------------------------------------------------
*--Define parameters
*
    PARAMETERS
*
*----define grid size and dimensions.
*----NX= cells in axial dir, NY = radial and nz = circumferential
*
  INTEGER NX=12, NY=6, NZ=30
*
*----seal dimensions in meters: RROT=rotor radius, RSTAT=stator radius,
*----RATRC=eccentricity ratio.
*
  REAL RROT=32.5E-3, RSTAT=52.586E-3
* REAL RROT=52.5E-3, RSTAT=52.586E-3
*
*----RL=length of seal: defined here in terms of L/D ratio.
*----set RL directly if needed.
*
  REAL LDRAT=0.4,  RL=LDRAT*2.*RROT, RY=1.5
*
*----RATRC=eccentricity ratio.
*
  REAL RATRC=0.00, CLEAR=RSTAT-RROT,  ECCEN=RATRC*CLEAR
*
*----define rotational parameters..
*
  REAL RPM=30400.0, PI=4.*ATAN(1.0), OMEGA=2.0*PI*RPM/60.0
*
*----define fluid properties for compressible flow.
*----AMU=dynamic viscosity, Pa-s, ANU=kinematic viscosity, m**2/s
*----DEN=density, K/m**3 WMOL=molecular weight, CP=specific heat J/Kg-K
*----TEMIN (K) and PRESIN (Pa) are upstream temp. and pres. values.
*
  REAL AMU=1.800E-5, TEMIN=650., PRESIN=1.52E6
  REAL WMOL=3.215, GAM=1.4, GASCON=8314./WMOL, CP=GAM*GASCON/(GAM-1.)
  REAL DEN1=PRESIN/GASCON/TEMIN, ANU=AMU/DEN1
*
*----entrance tke and eps based on reference inlet velocity, VVALUE
*----K1IN (m**2/s**2) and EPS (m**2/s**3) are tke and eps values.
*
  REAL VVALUE=200., K1IN=0.03*VVALUE**2, K12IN=SQRT(K1IN), CD=0.09
  REAL ET1IN=CD*K1IN*K12IN, E1IN=ET1IN*33./CLEAR
*
*----wvel specified inlet swirl
*
  REAL WVEL=RROT*OMEGA
*
    END
*
*------------------------------------------------------------------------------
*
```

```
    GEOMETRY
*
*----define grid type
*
  GRID 3D BFC
  L  NX ;  M  NY;  N  NZ
*
*----use the built in grid generator for this seal.
*
  GCSEAL 1 RL RROT RSTAT RY RATRC 0.0 0.0
*
    END
*
*-------------------------------------------------------------------------------
*
    PROBLEM_TYPE
*
*----define variables to be solved:
*
  SOLVE FLOW_C TURBULENCE HEAT
*
*---- activate following if rotordynamics is needed
*----*** PLEASE NOTE>>>> IMPORTANT
*----for a centered seal, i.e. with skew-symmetric coefficient matrix,
*----the circular whirl orbit method is the best.  use the following
*----command:
*
* ROTDYN  TRANSF  RPM CLEAR 0.1
*
*----use the perturbation method for nominally eccentric seals.
*
    END
*
*-------------------------------------------------------------------------------
*
PROPERTIES
*
*----constant viscosity and specific heat, perfect gas law for density.
*
  VISCOSITY CONSTANT ANU
  DENSITY GAS_LAW  PRESS=0.0  MOL_WT=WMOL
  SPECIFIC_HEAT CP
*
END
*
*-------------------------------------------------------------------------------
*
MODELS
*
*----default model for k-e used : Std. k-e with wall functions.
*
  TURBULENCE_MODEL KE
*
END
*
*-------------------------------------------------------------------------------
*
BOUNDARY_CONDITIONS
*----we choose to define each boundary region explicitly.
*----WARNING: if there are any boundary regions left unspecified, the
*---- preprocessor will impose a stationary, adiabatic wall on these regions,
*---- and will *not* inform the users.
*
*----stator wall
  WALL     1    L    M    M    1    N
   U=0.0  V=0.0  W=0.0
```

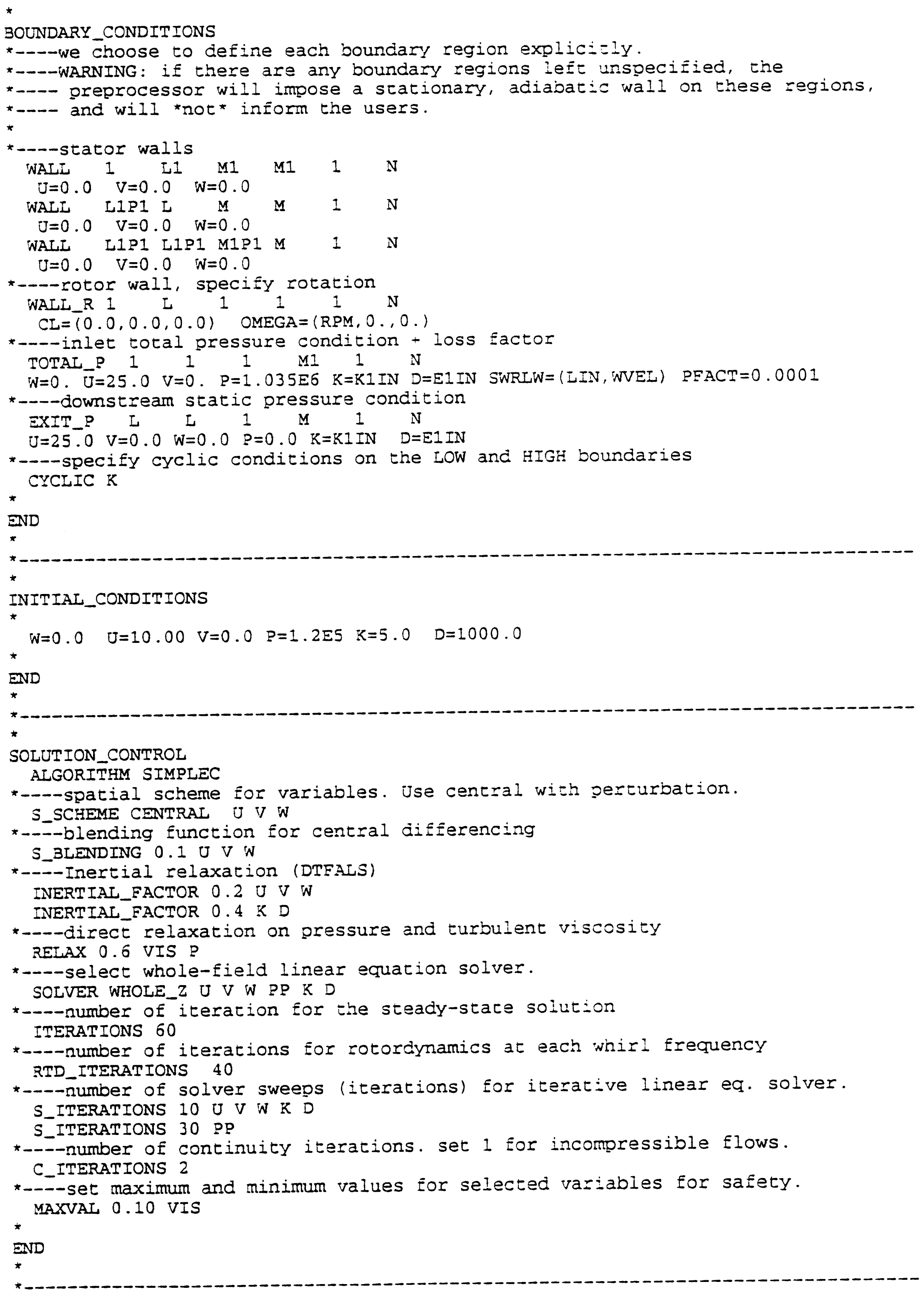
```
*----rotor wall, specify rotation
  WALL_R   1    L    1    1    1    N
   CL=(0.0,0.0,0.0) OMEGA=(RPM,0.0,0.0)
*----inlet total pressure condition + loss factor
  TOTAL_P  1    1    1    M    1    N
  W=0. U=16.5 V=0. P=1.52E6 T=650. K=K1IN D=E1IN SWRLW=(LIN,WVEL) PFACT=0.21
*----downstream static pressure condition
  EXIT_P   L    L    1    M    1    N
  U=16.5 V=0.0 W=0.0 P=0.65E6 K=K1IN  D=E1IN T=600
*
*----specify cyclic conditions on the LOW and HIGH boundaries
  CYCLIC K
*
END
*
*-------------------------------------------------------------------------------
*
INITIAL_CONDITIONS
*
  W=0.0  U=100.00 V=0.0 T=600. P=1.2E5 K=5.0  D=1000.0
*
END
*
*-------------------------------------------------------------------------------
*
SOLUTION_CONTROL
  ALGORITHM SIMPLEC
*----spatial scheme for variables.
  S_SCHEME CENTRAL  U V W
  S_SCHEME UPWIND  K D H
*----blending function for central differencing
  S_BLENDING 0.1 U V W
*----Inertial relaxation (DTFALS)
  INERTIAL_FACTOR 0.2 U V W
  INERTIAL_FACTOR 0.4 K D
  INERTIAL_FACTOR 0.1 H
*----direct relaxation on pressure and turbulent viscosity
  RELAX 0.6 VIS P
*----select whole-field linear equation solver.
  SOLVER WHOLE_Z U V W PP K D H
*----number of iteration for the steady-state solution
  ITERATIONS 75
*----number of iterations for rotordynamics at each whirl frequency
  RTD_ITERATIONS  70
*----number of solver sweeps (iterations) for iterative linear eq. solver.
  S_ITERATIONS 10 U V W K D H
  S_ITERATIONS 30 PP
*----number of continuity iterations. set 2-5 for compressible flows.
  C_ITERATIONS 2
*----set maximum and minimum values for selected variables for safety.
  MAXVAL 0.10 VIS
*
END
*
*----        ------------------------------------------------------------------
OUTPUT
*----print massflows through open areas
 M_FLOW ON
*----calculate and print rotor torque, power dissipation and pressure load
 RTORK ON
 RLOAD ON
 RPOWR ON
*----print wall info
*----if field print of additional variables is needed, add to this list.
 PRINTF WALL
END
```

Sample Input File for an Incompressible Flow Axial Step-Up Cylindrical Seal



```
TITLE  ' Incompressible flow seal, axial step-up '
*------------------------------------------------------------------------------
*******************************************************************************
*  This is a sample input file for a generic incompressible flow, seal with
*  an axial step-up in the stator diameter (const. rotor dia).
*******************************************************************************
* >>  All units below are in SI. <<
*
*  PLEASE CHECK THE USERS MANUAL IF ADDITIONAL INFORMATION IS NEEDED.
*
*  Contact Mahesh Athavale at mma@cfdl.cfdrc.com
*                   or call at (205)536-6576
*  if you need something beyond the material here or as a last resort.
*
*------------------------------------------------------------------------------
*--Define parameters
*
     PARAMETERS
*
*----define grid size and dimensions.
*----NX= cells in axial dir, NY = radial and nz = circumferential
*
  INTEGER NX=20, NY=12, NZ=24
*
*----define sizes for step.
*----L1=cells in axial dir. before step, L2=cells after step (NX=L1+L2)
*----M1=cells in smaller radial gap, M2=cells in larger gap. (NY = M2)
*
  INTEGER L1=10, L2=10, L1P1=L1+1
  INTEGER M1=6,  M2=12, M1P1=M1+1
*
*----Seal dimensions in meters
*----RROT=rotor radius, RL=total seal length
*----CSMALL=smaller clearance, CLARGE=larger clearance
*----RSTAT1=smaller stator radius, RSTAT2=larger stator radius
*----RATRC=eccentricity ratio = eccentricity/min. clearance
*
  REAL RROT=7.98E-2/2.,  RL=4.32E-2
  REAL CSMALL=2.967E-2,  CLARGE=1.5*CSMALL
* REAL CSMALL=2.967E-4,  CLARGE=1.5*CSMALL
  REAL RSTAT1=RROT+CSMALL, RSTAT2=RROT+CLARGE
  REAL RATRC=0.00, CLEAR=RSTAT1-RROT,  ECCEN=RATRC*CLEAR
*
*----seal lengths:  XL1=axial length before step, XL2= after step.
*----RX1=clustering parameter to cluster grid near step-up wall.
*
  REAL XL1=RL, XL2=XL1,  RY1=1.5,  RX1=1.5, RY2=1.5
*
*
*----define rotational parameters..
*
  REAL RPM=3600.0, PI=4.*ATAN(1.0), OMEGA=2.0*PI*RPM/60.0
*
*----AMU=dynamic viscosity, Pa-s, ANU=kinematic viscosity, m**2/s
*----DEN=density, K/m**3
*
  REAL AMU=1.245E-3, DEN=1000.0, ANU=AMU/DEN
*
*----entrance tke and eps based on reference inlet velocity, VVALUE
*----K1IN (m**2/s**2) and EPS (m**2/s**3) are tke and eps values.
*----you could assign these values directly to K1IN and E1IN if known.
*
  REAL VVALUE=25.0, K1IN=0.03*VVALUE**2, K12IN=SQRT(K1IN), CD=0.09
  REAL ET1IN=CD*K1IN*K12IN, E1IN=ET1IN*2./CLEAR
*
*----wvel specified inlet swirl
```

```
*
  REAL WVEL=RROT*OMEGA
*
    END
*
*--------------------------------------------------------------------------------
*
    GEOMETRY
*
*----define grid type
*
  GRID 3D BFC
  L  NX ;  M  NY;  N  NZ
*
*----use the built in grid generator for this seal.
*
  GCSEAL 3 L1 L2 M1 M2 XL1 XL2 RROT RSTAT1 RSTAT2 RX1 RY1 RY2 RATRC
*
*----blockage to generate the step-up on the stator.
*
  BLOCK  1    L1   M1P1  NY  1   NZ
*
    END
*
*--------------------------------------------------------------------------------
*
    PROBLEM_TYPE
*
*----define variables to be solved:
*
  SOLVE FLOW TURBULENCE
*
*---- activate following if rotordynamics is needed
*----*** PLEASE NOTE>>>> IMPORTANT
*----for a centered seal, i.e. with skew-symmetric coefficient matrix,
*----the circular whirl orbit method is the best.  use the following
*----command:
*
* ROTDYN  TRANSF  RPM CLEAR 0.1
*
*----use the perturbation method for nominally eccentric seals.
*
    END
*
*--------------------------------------------------------------------------------
*
PROPERTIES
*
*----constant viscosity and  density
*
  VISCOSITY CONSTANT ANU
  DENSITY CONSTANT DEN
*
END
*
*--------------------------------------------------------------------------------
*
MODELS
*
*----default model for k-e used : Std. k-e with wall functions.
*
  TURBULENCE_MODEL KE
*
END
*
*--------------------------------------------------------------------------------
```



```
*
BOUNDARY_CONDITIONS
*----we choose to define each boundary region explicitly.
*----WARNING: if there are any boundary regions left unspecified, the
*---- preprocessor will impose a stationary, adiabatic wall on these regions,
*---- and will *not* inform the users.
*
*----stator walls
  WALL    1    L1   M1   M1   1    N
   U=0.0  V=0.0  W=0.0
  WALL    L1P1 L    M    M    1    N
   U=0.0  V=0.0  W=0.0
  WALL    L1P1 L1P1 M1P1 M    1    N
   U=0.0  V=0.0  W=0.0
*----rotor wall, specify rotation
  WALL_R 1    L    1    1    1    N
   CL=(0.0,0.0,0.0)  OMEGA=(RPM,0.,0.)
*----inlet total pressure condition + loss factor
  TOTAL_P  1    1    1    M1   1    N
  W=0. U=25.0 V=0. P=1.035E6 K=K1IN D=E1IN SWRLW=(LIN,WVEL) PFACT=0.0001
*----downstream static pressure condition
  EXIT_P   L    L    1    M    1    N
  U=25.0 V=0.0 W=0.0 P=0.0 K=K1IN  D=E1IN
*----specify cyclic conditions on the LOW and HIGH boundaries
  CYCLIC K
*
END
*
*----------------------------------------------------------------------------
*
INITIAL_CONDITIONS
*
  W=0.0  U=10.00 V=0.0 P=1.2E5 K=5.0  D=1000.0
*
END
*
*----------------------------------------------------------------------------
*
SOLUTION_CONTROL
  ALGORITHM SIMPLEC
*----spatial scheme for variables. Use central with perturbation.
  S_SCHEME CENTRAL  U V W
*----blending function for central differencing
  S_BLENDING 0.1 U V W
*----Inertial relaxation (DTFALS)
  INERTIAL_FACTOR 0.2 U V W
  INERTIAL_FACTOR 0.4 K D
*----direct relaxation on pressure and turbulent viscosity
  RELAX 0.6 VIS P
*----select whole-field linear equation solver.
  SOLVER WHOLE_Z U V W PP K D
*----number of iteration for the steady-state solution
  ITERATIONS 60
*----number of iterations for rotordynamics at each whirl frequency
  RTD_ITERATIONS  40
*----number of solver sweeps (iterations) for iterative linear eq. solver.
  S_ITERATIONS 10 U V W K D
  S_ITERATIONS 30 PP
*----number of continuity iterations. set 1 for incompressible flows.
  C_ITERATIONS 2
*----set maximum and minimum values for selected variables for safety.
  MAXVAL 0.10 VIS
*
END
*
*----------------------------------------------------------------------------
```

```
OUTPUT
*----print massflows through open areas
 M_FLOW ON
*----calculate and print rotor torque, power dissipation and pressure load
 RTORK ON
 RLOAD ON
 RPOWR ON
*----print wall info
*----if field print of additional variables is needed, add to this list.
 PRINTF WALL
END
```

Sample Input File for an Incompressible Flow Axial Step-Down Cylindrical Seal

```
TITLE  ' Incompressible flow seal, axial step-down '
*------------------------------------------------------------------------------
*******************************************************************************
*  This is a sample input file for a generic incompressible flow,
*  seal with an axial step-down in stator diameter ( const. rotor dia.)
*******************************************************************************
* >>  All units below are in SI.  <<
*
*  PLEASE CHECK THE USERS MANUAL IF ADDITIONAL INFORMATION IS NEEDED.
*
*  Contact Mahesh Athavale at mma@cfdl.cfdrc.com
*                 or call at (205)536-6576
*  if you need something beyond the material here or as a last resort.
*
*------------------------------------------------------------------------------
*--Define parameters
*
     PARAMETERS
*
*----define grid size and dimensions.
*----NX= cells in axial dir, NY = radial and nz = circumferential
*
  INTEGER NX=20, NY=12, NZ=24
*
*----define sizes for step.
*----L1=cells in axial dir. before step, L2=cells after step (NX=L1+L2)
*----M1=cells in smaller radial gap, M2=cells in larger gap. (NY = M2)
*
  INTEGER L1=10, L2=10, L1P1=L1+1
  INTEGER M1=6,  M2=12, M1P1=M1+1
*
*----Seal dimensions in meters
*----RROT=rotor radius, RL=total seal length
*----CSMALL=smaller clearance, CLARGE=larger clearance
*----RSTAT1=smaller stator radius, RSTAT2=larger stator radius
*----RATRC=eccentricity ratio = eccentricity/min. clearance
*
  REAL RROT=7.98E-2/2.
*  REAL CSMALL=2.967E-4,  CLARGE=2.0*CSMALL
  REAL CSMALL=2.967E-2,  CLARGE=2.0*CSMALL
  REAL RSTAT1=RROT+CSMALL, RSTAT2=RROT+CLARGE
  REAL RATRC=0.00, CLEAR=RSTAT1-RROT,  ECCEN=RATRC*CLEAR
*
*----eal lengths:  XL1=axial length before step, XL2= after step.
*----RX1=clustering parameter to cluster grid near step-down wall.
*
  REAL XL1=4.32E-2, XL2=XL1,  RY1=1.5,  RX1=1.4, RY2=1.5
*
*----define rotational parameters..
*
  REAL RPM=3600.0, PI=4.*ATAN(1.0), OMEGA=2.0*PI*RPM/60.0
*
*----define physical parameters
*
*----AMU=dynamic viscosity, Pa-s, ANU=kinematic viscosity, m**2/s
*----DEN=density, K/m**3
*
  REAL AMU=1.245E-3, DEN=1000.0, ANU=AMU/DEN
*
*-entrance tke and eps based on reference inlet ---velocity, VVALUE
*----K1IN (m**2/s**2) and EPS (m**2/s**3) are tke and eps values.
*----you could assign these values directly to K1IN and E1IN if known.
*
  REAL VVALUE=25.0, K1IN=0.03*VVALUE**2, K12IN=SQRT(K1IN), CD=0.09
  REAL ET1IN=CD*K1IN*K12IN, E1IN=ET1IN*2./CLEAR
*
```

```
*----wvel specified inlet swirl
*
  REAL WVEL=RROT*OMEGA
*
    END
*
*-------------------------------------------------------------------------------
*
    GEOMETRY
*
*----define grid type
*
  GRID 3D BFC
  L  NX ;  M  NY;  N  NZ
*
*----use the built in grid generator for this seal.
*
  GCSEAL 2 L1 L2 M1 M2 XL1 XL2 RROT RSTAT1 RSTAT2 RX1 RY1 RY2 RATRC
*
*----impose blocked cells
*
  BLOCK  L1P1  L  M1P1  M  1  N
*
    END
*
*-------------------------------------------------------------------------------
*
    PROBLEM_TYPE
*
*----define variables to be solved:
*
  SOLVE FLOW TURBULENCE
*
*---- activate following if rotordynamics is needed
*----*** PLEASE NOTE>>>> IMPORTANT
*----for a centered seal, i.e. with skew-symmetric coefficient matrix,
*----the circular whirl orbit method is the best.  use the following
*----command:
*
* ROTDYN  TRANSF  RPM CLEAR 0.1
*
*----use the perturbation method for nominally eccentric seals.
*
    END
*
*-------------------------------------------------------------------------------
*
PROPERTIES
*
*----constant viscosity and  density
*
  VISCOSITY CONSTANT ANU
  DENSITY CONSTANT DEN
*
END
*
*-------------------------------------------------------------------------------
*
MODELS
*
*----default model for k-e used : Std. k-e with wall functions.
*
  TURBULENCE_MODEL KE
*
END
*
```

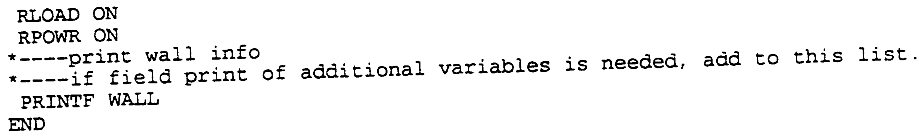
```
*-------------------------------------------------------------------------------
*
BOUNDARY_CONDITIONS
*----we choose to define each boundary region explicitly.
*----WARNING: if there are any boundary regions left unspecified, the
*---- preprocessor will impose a stationary, adiabatic wall on these regions,
*---- and will *not* inform the users.
*
*----stator walls
  WALL   1    L1  M     M    1    N
   U=0.0  V=0.0  W=0.0
  WALL   L1P1 L    M1   M1   1    N
   U=0.0  V=0.0  W=0.0
  WALL   L1   L1   M1P1 M    1    N
   U=0.0  V=0.0  W=0.0
*----rotor wall
  WALL_R   1    L    1    1    1    N
   CL=(0.0,0.0,0.0) OMEGA=(RPM,0.0,0.0)
*----inlet total pressure condition + loss factor
  TOTAL_P  1    1    1    M    1    N
   W=0. U=25.0 V=0. P=1.035E6 K=K1IN D=E1IN SWRLW=(LIN,WVEL) PFACT=0.0001
*----downstream static pressure condition
  EXIT_P   L    L    1    M1   1    N
   W=0. U=25.0 V=0. P=0.00 K=K1IN D=E1IN
*
*----specify cyclic conditions on the LOW and HIGH boundaries
  CYCLIC K
*
END
*
*-------------------------------------------------------------------------------
*
INITIAL_CONDITIONS
*
  W=0.0  U=10.00 V=0.0 P=1.2E5 K=5.0  D=1000.0
*
END
*
*-------------------------------------------------------------------------------
*
SOLUTION_CONTROL
  ALGORITHM SIMPLEC
*----spatial scheme for variables. Use CENTRAL for perturbation method.
  S_SCHEME CENTRAL  U V W
*----blending function for central differencing
  S_BLENDING 0.1 U V W
*----Inertial relaxation (DTFALS)
  INERTIAL_FACTOR 0.3 U V W
  INERTIAL_FACTOR 0.6 K D
*----direct relaxation on pressure and turbulent viscosity
  RELAX 0.5 VIS P
*----select whole-field linear equation solver.
  SOLVER WHOLE_Z U V W PP K D
*----number of iteration for the steady-state solution
  ITERATIONS 40
*----number of iterations for rotordynamics at each whirl frequency
  RTD_ITERATIONS  30
*----number of solver sweeps (iterations) for iterative linear eq. solver.
  S_ITERATIONS 10 U V W K D
  S_ITERATIONS 30 PP
*----number of continuity iterations. set 1 for incompressible flows.
  C_ITERATIONS 2
*----set maximum and minimum values for selected variables for safety.
  MAXVAL 10.0 VIS
*
END
```

```
*
*-------------------------------------------------------------------------------
OUTPUT
*----print massflows through open areas
 M_FLOW ON
*----calculate and print rotor torque, power dissipation and pressure load
 RTORK ON
 RLOAD ON
 RPOWR ON
*----print wall info
*----if field print of additional variables is needed, add to this list.
 PRINTF WALL
END
```

Sample Input File for an Incompressible Flow, Tapered-Land Cylindrical Seal

```
TITLE  ' Incompressible flow seal, tapered land seal '
*-------------------------------------------------------------------------------
********************************************************************************
*  This is a sample input file for a generic incompressible flow, seal with
*  tapered land (const. rotor dia).
********************************************************************************
* >>  All units below are in SI.  <<
*
*  PLEASE CHECK THE USERS MANUAL IF ADDITIONAL INFORMATION IS NEEDED.
*
*  Contact Mahesh Athavale at mma@cfdl.cfdrc.com
*                  or call at (205)536-6576
*  if you need something beyond the material here or as a last resort.
*
*-------------------------------------------------------------------------------
*--Define parameters
*
     PARAMETERS
*
*----define grid size and dimensions.
*----NX= cells in axial dir, NY = radial and nz = circumferential
*
* INTEGER NX=12, NY=6, NZ=30
  INTEGER NX=16, NY=6, NZ=30
*
*  define limits in axial direction for tapered land.
*  L1=axial cells in tapered portion, L2=axial cells in land section
*   L2=0 for no land,  L1+L2 must match NX.
*
* INTEGER L1=NX, L2=0
  INTEGER L1=12, L2=4
*
*  seal dimensions in meters:
*      RROT=rotor radius, RSTAT1=smaller stator radius,
*      RSTAT2=larger stator radius.
*      RATRC=eccentricity ratio.
*
*  REAL RROT=80.0E-3, RSTAT1=80.36E-3, RSTAT2=80.72E-3
  REAL RROT=50.0E-3, RSTAT1=70.36E-3, RSTAT2=80.72E-3
  REAL RATRC=0.00, CLEAR=RSTAT1-RROT,  ECCEN=RATRC*CLEAR
*
*  length of seal:
*  set XL1 directly.  XL2 is the axial length of the land.
*
* REAL XL1=40.E-3, XL2=0.0, RY1=1.0
  REAL XL1=40.E-3, XL2=20.E-3, RY1=1.5
*
*  RATRC=eccentricity ratio.
*
  REAL RATRC=0.00, CLEAR=RSTAT1-RROT,  ECCEN=RATRC*CLEAR
*
*----define rotational parameters..
*
  REAL RPM=4000.0, PI=4.*ATAN(1.0), OMEGA=2.0*PI*RPM/60.0
*
*  define fluid properties for compressible flow.
*  AMU=dynamic viscosity, Pa-s, ANU=kinematic viscosity, m**2/s
*  DEN=density, K/m**3
*
  REAL AMU=1.000E-3, DEN=1000.0, ANU=AMU/DEN
*
*----entrance tke and eps based on reference inlet velocity, VVALUE
*----K1IN (m**2/s**2) and EPS (m**2/s**3) are tke and eps values.
*----you could assign these values directly to K1IN and E1IN if known.
*
  REAL VVALUE=25.0, K1IN=0.03*VVALUE**2, K12IN=SQRT(K1IN), CD=0.09
  REAL ET1IN=CD*K1IN*K12IN, E1IN=ET1IN*2./CLEAR
*
*----wvel specified inlet swirl
*
  REAL WVEL=RROT*OMEGA
*
    END
*
*--------------------------------------------------------------------------------
*
    GEOMETRY
*
*----define grid type
*
  GRID 3D BFC
  L  NX ;  M  NY;  N  NZ
*
*----use the built in grid generator for this seal.
*
  GCSEAL 4 L1 L2 XL1 XL2  RROT RSTAT1 RSTAT2 RY1 RATRC
*
    END
*
*--------------------------------------------------------------------------------
*
    PROBLEM_TYPE
*
*----define variables to be solved:
*
  SOLVE FLOW TURBULENCE
*
*---- activate following if rotordynamics is needed
*----*** PLEASE NOTE>>>> IMPORTANT
*----for a centered seal, i.e. with skew-symmetric coefficient matrix,
*----the circular whirl orbit method is the best.  use the following
*----command:
*
* ROTDYN  TRANSF  RPM CLEAR 0.1
*
*----use the perturbation method for nominally eccentric seals.
*
    END
*
*--------------------------------------------------------------------------------
*
PROPERTIES
*
*----constant viscosity and  density
*
  VISCOSITY CONSTANT ANU
  DENSITY CONSTANT DEN
*
END
*
*--------------------------------------------------------------------------------
*
MODELS
*
*----default model for k-e used : Std. k-e with wall functions.
*
  TURBULENCE_MODEL KE
*
END
*
*--------------------------------------------------------------------------------
*
```

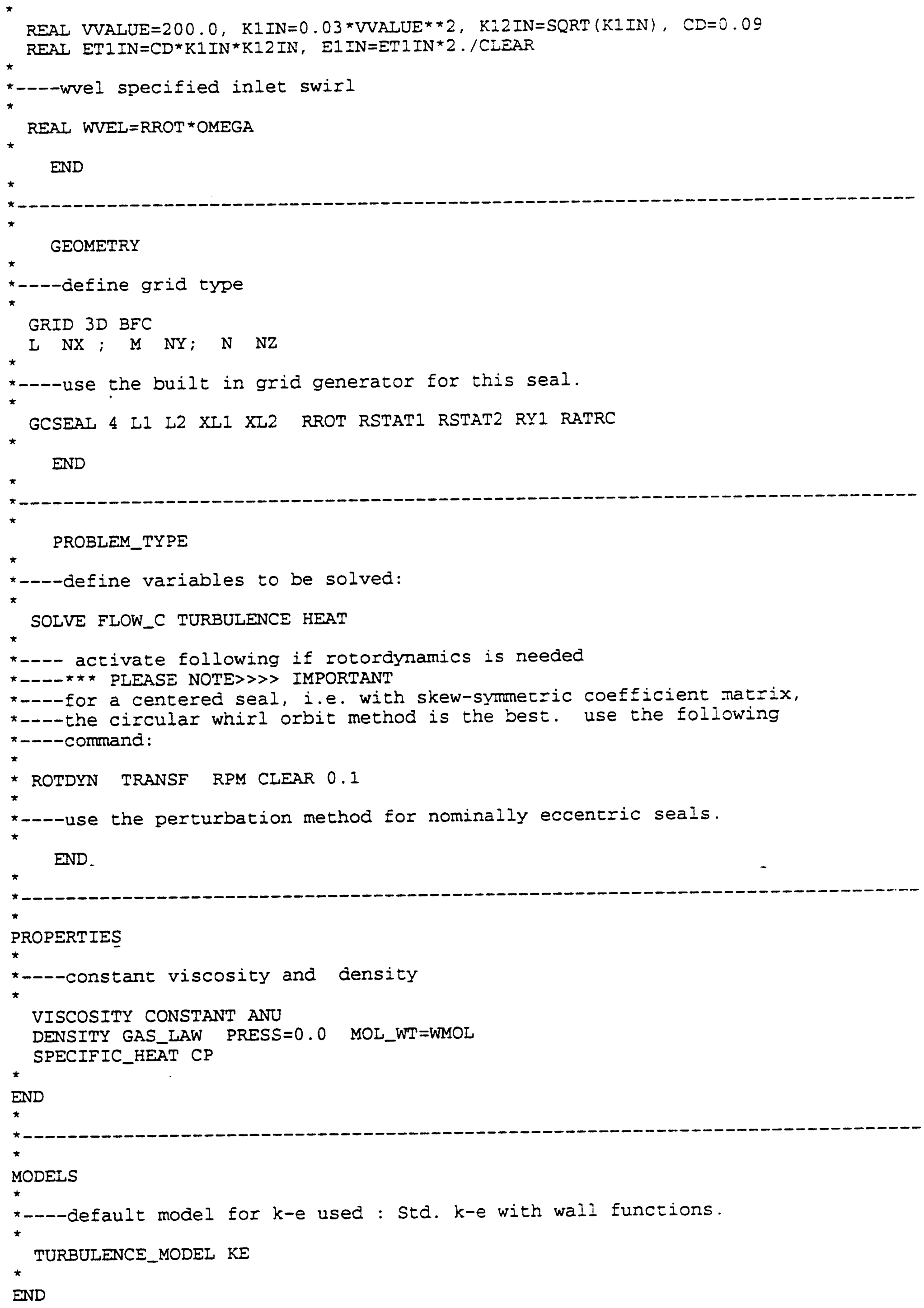
```
BOUNDARY_CONDITIONS
*----we choose to define each boundary region explicitly.
*----WARNING: if there are any boundary regions left unspecified, the
*---- preprocessor will impose a stationary, adiabatic wall on these regions,
*---- and will *not* inform the users.
*
*----stator wall
  WALL   1    L    M    M    1    N
   U=0.0  V=0.0  W=0.0
*----rotor wall, specify rotation
  WALL_R 1    L    1    1    1    N
   CL=(0.0,0.0,0.0)  OMEGA=(RPM,0.,0.)
*----inlet total pressure condition + loss factor
  TOTAL_P  1    1    1    M    1    N
  W=0. U=25.0 V=0. P=1.035E6 K=K1IN D=E1IN SWRLW=(LIN,WVEL) PFACT=0.0001
*----downstream static pressure condition
  EXIT_P   L    L    1    M    1    N
  U=25.0 V=0.0 W=0.0 P=0.0 K=K1IN  D=E1IN
*----specify cyclic conditions on the LOW and HIGH boundaries
  CYCLIC K
*
END
*
*---------------------------------------------------------------------------
*
INITIAL_CONDITIONS
*
  W=0.0  U=20.00 V=0.0 P=1.2E5 K=5.0  D=1000.0
*
END
*
*---------------------------------------------------------------------------
*
SOLUTION_CONTROL
  ALGORITHM SIMPLEC
*----spatial scheme for variables.  Use CENTRAL with perturbation model.
  S_SCHEME CENTRAL  U V W
*----blending function for central differencing
  S_BLENDING 0.1 U V W
*----Inertial relaxation (DTFALS)
  INERTIAL_FACTOR 0.2 U V W
  INERTIAL_FACTOR 0.4 K D
*----direct relaxation on pressure and turbulent viscosity
  RELAX 0.6 VIS P
*----select whole-field linear equation solver.
  SOLVER WHOLE_Z U V W PP K D
*----number of iteration for the steady-state solution
  ITERATIONS 40
*----number of iterations for rotordynamics at each whirl frequency
  RTD_ITERATIONS  40
*----number of solver sweeps (iterations) for iterative linear eq. solver.
  S_ITERATIONS 10 U V W K D
  S_ITERATIONS 30 PP
*----number of continuity iterations. set 1 for incompressible flows.
  C_ITERATIONS 3
*----set maximum and minimum values for selected variables for safety.
  MAXVAL 0.10 VIS
*
END
*
*---------------------------------------------------------------------------
OUTPUT
*----print massflows through open areas
 M_FLOW ON
*----calculate and print rotor torque, power dissipation and pressure load
 RTORK ON
```



```
 RLOAD ON
 RPOWR ON
*----print wall info
*----if field print of additional variables is needed, add to this list.
 PRINTF WALL
END
```

Sample Input File for a Compressible Flow, Tapered-Land Cylindrical Seal

```
TITLE  ' Compressible flow seal, tapered land seal '
*-----------------------------------------------------------------------------
******************************************************************************
*  This is a sample input file for a generic compressible flow, seal with
*  tapered land (const. rotor dia).
******************************************************************************
* >>  All units below are in SI. <<
*
*  PLEASE CHECK THE USERS MANUAL IF ADDITIONAL INFORMATION IS NEEDED.
*
*  Contact Mahesh Athavale at mma@cfdl.cfdrc.com
*                  or call at (205)536-6576
*  if you need something beyond the material here or as a last resort.
*
*-----------------------------------------------------------------------------
*--Define parameters
*
    PARAMETERS
*
*----define grid size and dimensions.
*----NX= cells in axial dir, NY = radial and nz = circumferential
*
* INTEGER NX=16, NY=12, NZ=24
  INTEGER NX=20, NY=12, NZ=24
*
*  define limits in axial direction for tapered land.
*  L1=axial cells in tapered portion, L2=axial cells in land section
*   L2=0 for no land,  L1+L2 must match NX.
*
* INTEGER L1=NX, L2=0
  INTEGER L1=12, L2=8
*
*  seal dimensions in meters:
*      RROT=rotor radius, RSTAT1=smaller stator radius,
*      RSTAT2=larger stator radius.
*      RATRC=eccentricity ratio.
*
* REAL RROT=32.5E-3, RSTAT1=32.586E-3, RSTAT2=32.672E-3
  REAL RROT=32.5E-3, RSTAT1=42.586E-3, RSTAT2=52.672E-3
*
*  length of seal: defined here in terms of L/D ratio.
*  set XL1 directly if needed.  XL2 is the axial length of the land.
*
* REAL LDRAT=0.2,  XL1=LDRAT*2.*RROT, XL2=0.0, RY1=1.0
  REAL LDRAT=0.2,  XL1=LDRAT*2.*RROT, XL2=30.E-3, RY1=1.5
*
*  RATRC=eccentricity ratio.
*
  REAL RATRC=0.50, CLEAR=RSTAT1-RROT,  ECCEN=RATRC*CLEAR
*
*----define rotational parameters..
*
  REAL RPM=30400.0, PI=4.*ATAN(1.0), OMEGA=2.0*PI*RPM/60.0
*
*  define fluid properties for compressible flow.
*  AMU=dynamic viscosity, Pa-s, ANU=kinematic viscosity, m**2/s
*  DEN=density, K/m**3 WMOL=molecular weight, CP=specific heat J/Kg-K
*  TEMIN (K) and PRESIN (Pa) are upstream temp. and pres. values.
*
  REAL AMU=1.800E-5,  TEMIN=600., PRESIN=1.52E6
  REAL WMOL=3.215, GAM=1.4, GASCON=8314./WMOL, CP=GAM*GASCON/(GAM-1.)
  REAL DEN1=PRESIN/GASCON/TEMIN, ANU=AMU/DEN1
*
*----entrance tke and eps based on reference inlet velocity, VVALUE
*----K1IN (m**2/s**2) and EPS (m**2/s**3) are tke and eps values.
*----you could assign these values directly to K1IN and E1IN if known.
```

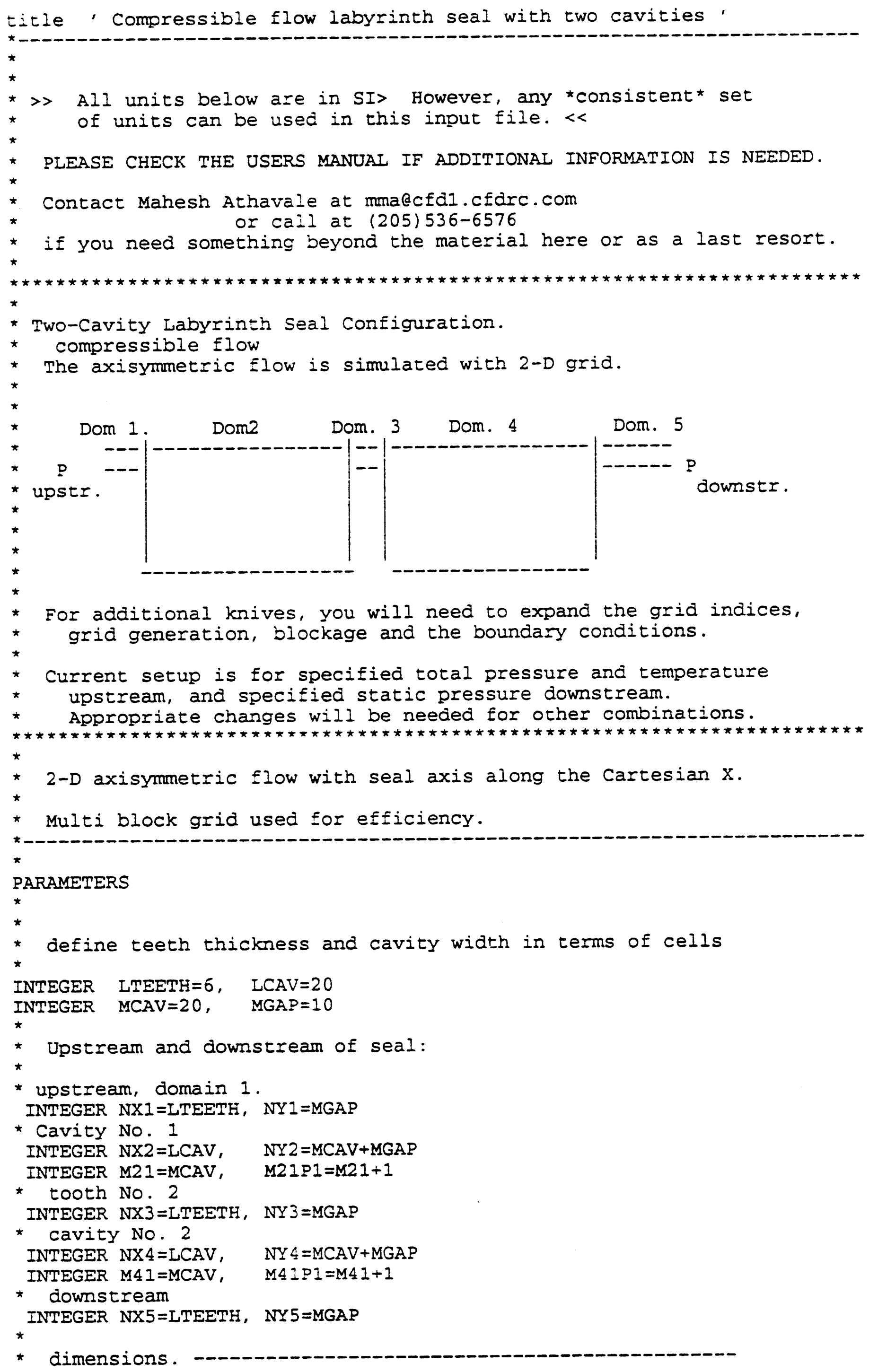


```
*
  REAL VVALUE=200.0, K1IN=0.03*VVALUE**2, K12IN=SQRT(K1IN), CD=0.09
  REAL ET1IN=CD*K1IN*K12IN, E1IN=ET1IN*2./CLEAR
*
*----wvel specified inlet swirl
*
  REAL WVEL=RROT*OMEGA
*
    END
*
*----------------------------------------------------------------------------
*
    GEOMETRY
*
*----define grid type
*
  GRID 3D BFC
  L  NX ;  M  NY;  N  NZ
*
*----use the built in grid generator for this seal.
*
  GCSEAL 4 L1 L2 XL1 XL2  RROT RSTAT1 RSTAT2 RY1 RATRC
*
    END
*
*----------------------------------------------------------------------------
*
    PROBLEM_TYPE
*
*----define variables to be solved:
*
  SOLVE FLOW_C TURBULENCE HEAT
*
*---- activate following if rotordynamics is needed
*----*** PLEASE NOTE>>>> IMPORTANT
*----for a centered seal, i.e. with skew-symmetric coefficient matrix,
*----the circular whirl orbit method is the best.  use the following
*----command:
*
* ROTDYN  TRANSF  RPM CLEAR 0.1
*
*----use the perturbation method for nominally eccentric seals.
*
    END
*
*----------------------------------------------------------------------------
*
PROPERTIES
*
*----constant viscosity and  density
*
  VISCOSITY CONSTANT ANU
  DENSITY GAS_LAW  PRESS=0.0  MOL_WT=WMOL
  SPECIFIC_HEAT CP
*
END
*
*----------------------------------------------------------------------------
*
MODELS
*
*----default model for k-e used : Std. k-e with wall functions.
*
  TURBULENCE_MODEL KE
*
END
```

```
*
*-------------------------------------------------------------------------------
*
BOUNDARY_CONDITIONS
*----we choose to define each boundary region explicitly.
*----WARNING: if there are any boundary regions left unspecified, the
*---- preprocessor will impose a stationary, adiabatic wall on these regions,
*---- and will *not* inform the users.
*
*----stator wall
  WALL   1    L   M   M   1    N
   U=0.0  V=0.0  W=0.0
*----rotor wall, specify rotation
  WALL_R 1    L   1    1    1    N
   CL=(0.0,0.0,0.0)  OMEGA=(RPM,0.,0.)
*----inlet total pressure condition + loss factor
  TOTAL_P  1    1    1    M   1    N
  W=0. U=25.0 V=0. P=1.035E6 T=700. K=K1IN D=E1IN SWRLW=(LIN,WVEL) PFACT=0.0001
*----downstream static pressure condition
  EXIT_P   L    L    1    M    1    N
  U=25.0 V=0.0 W=0.0 P=0.1E5 K=K1IN  D=E1IN T=600.0
*----specify cyclic conditions on the LOW and HIGH boundaries
  CYCLIC K
*
END
*
*-------------------------------------------------------------------------------
*
INITIAL_CONDITIONS
*
  W=0.0  U=200.00 V=0.0 P=1.2E5 K=5.0  D=1000.0 T=TEMIN
*
END
*
*-------------------------------------------------------------------------------
*
SOLUTION_CONTROL
  ALGORITHM SIMPLEC
*----spatial scheme for variables.
  S_SCHEME CENTRAL  U V W
  S_SCHEME UPWIND  K D H
*----blending function for central differencing
  S_BLENDING 0.1 U V W
*----Inertial relaxation (DTFALS)
  INERTIAL_FACTOR 0.2 U V W H
  INERTIAL_FACTOR 0.4 K D
*----direct relaxation on pressure and turbulent viscosity
  RELAX 0.6 VIS P
*----select whole-field linear equation solver.
  SOLVER WHOLE_Z U V W PP K D H
*----number of iteration for the steady-state solution
  ITERATIONS 60
*----number of iterations for rotordynamics at each whirl frequency
  RTD_ITERATIONS  40
*----number of solver sweeps (iterations) for iterative linear eq. solver.
  S_ITERATIONS 10 U V W K D H
  S_ITERATIONS 30 PP
*----number of continuity iterations. set 1 for incompressible flows.
  C_ITERATIONS 3
*----set maximum and minimum values for selected variables for safety.
  MAXVAL 0.10 VIS
*
END
*
*-------------------------------------------------------------------------------
OUTPUT
```

```
*----print massflows through open areas
 M_FLOW ON
*----calculate and print rotor torque, power dissipation and pressure load
 RTORK ON
 RLOAD ON
 RPOWR ON
*----print wall info
*----if field print of additional variables is needed, add to this list.
 PRINTF WALL
END
```

Sample Input File for Axisymmetric Compressible Flow in a 2-Cavity Labyrinth Seal (Multi-domain version)



```
title  ' Compressible flow labyrinth seal with two cavities '
*-----------------------------------------------------------------------
*
*
* >>  All units below are in SI>  However, any *consistent* set
*     of units can be used in this input file. <<
*
*  PLEASE CHECK THE USERS MANUAL IF ADDITIONAL INFORMATION IS NEEDED.
*
*  Contact Mahesh Athavale at mma@cfdl.cfdrc.com
*                  or call at (205)536-6576
*  if you need something beyond the material here or as a last resort.
*
************************************************************************
*
* Two-Cavity Labyrinth Seal Configuration.
*   compressible flow
*  The axisymmetric flow is simulated with 2-D grid.
*
*
*     Dom 1.     Dom2      Dom. 3    Dom. 4       Dom. 5
*       ---|----------------|--|-----------------|------
*   P   ---|                |--|                 |------ P
* upstr.   |                |  |                 |        downstr.
*          |                |  |                 |
*          |                |  |                 |
*          |                |  |                 |
*          -------------------  -----------------
*
*  For additional knives, you will need to expand the grid indices,
*    grid generation, blockage and the boundary conditions.
*
*  Current setup is for specified total pressure and temperature
*    upstream, and specified static pressure downstream.
*    Appropriate changes will be needed for other combinations.
************************************************************************
*
*  2-D axisymmetric flow with seal axis along the Cartesian X.
*
*  Multi block grid used for efficiency.
*-----------------------------------------------------------------------
*
PARAMETERS
*
*
*  define teeth thickness and cavity width in terms of cells
*
INTEGER  LTEETH=6,  LCAV=20
INTEGER  MCAV=20,   MGAP=10
*
*  Upstream and downstream of seal:
*
* upstream, domain 1.
 INTEGER NX1=LTEETH, NY1=MGAP
* Cavity No. 1
 INTEGER NX2=LCAV,   NY2=MCAV+MGAP
 INTEGER M21=MCAV,   M21P1=M21+1
*  tooth No. 2
 INTEGER NX3=LTEETH, NY3=MGAP
*  cavity No. 2
 INTEGER NX4=LCAV,   NY4=MCAV+MGAP
 INTEGER M41=MCAV,   M41P1=M41+1
*  downstream
 INTEGER NX5=LTEETH, NY5=MGAP
*
*  dimensions. ---------------------------------------------
*  axial dimensions:  xup, xdown, xcav and xteeth
*  fill up these with dimensiona for your case.
*
 REAL XUP=2.E-3, XDOWN=10.E-3,  XTEETH=2.E-3,  XCAV=10.E-3
*
*  define locations for grid generation
*
 REAL XMIN=0.0,  X1=XMIN+XUP,  X2=X1+XCAV,  X3=X2+XTEETH
 REAL X4=X3+XCAV,  X5=X4+XDOWN
*
*  radial direction, ycav and ygap
*
 REAL YCAV=10.E-3,  YGAP=0.7E-3
*
 REAL YMIN=25.E-3,  Y1=YMIN+YCAV,  Y2=Y1+YGAP
*
*  physical parameters.
*  rpm=rotor speed in rpm.,  densit=density pupst=upstrem pressure
*  pdown=downstream pressure. tupst=upstrem temperature
* pressure in Pa, temp in Kelvin.
*  densit=density kg/m**3, vislam=kinematic viscosity, m**2/s
*
 REAL RPM=4000.,  PI=3.14159, RPS=RPM/60., OMX=2.*PI*RPS
 REAL PUPST=1.4E5,  PDOWN=1.0E5, TUPST=400., TDOWN=TUPST
 REAL DENSIT=1.2,  VISLAM=1.E-5
 REAL UINL=200.0
*
*
END
*------------------------------------------------------------------------
*
GEOMETRY
 GRID 2D BFC  AXISYMMETRIC
*
* DOMAIN 1
*
 L  NX1
 M  NY1
*
 ILINE 1   1  LP1  (XMIN,Y1)  (X1,Y1)    1.0
 ILINE MP1 1  LP1  (XMIN,Y2)  (X1,Y2)    1.0
 JLINE 1   1  MP1  (XMIN,Y1)  (XMIN,Y2)  1.3  SYMM
 JLINE LP1 1  MP1  (X1,Y1)    (X1,Y2)    1.3  SYMM
*
 FILSRF  1   LP1   1   MP1
*
*---------
*
 DOMAIN 2
*
 L  NX2
 M  NY2
*
 ILINE  1      1  LP1  (X1,YMIN) (X2,YMIN)  1.3  SYMM
 ILINE  M21P1 1  LP1  (X1,Y1)   (X2,Y1)    1.3  SYMM
 ILINE  MP1    1  LP1  (X1,Y2)   (X2,Y2)    1.3  SYMM
*
 JLINE  1    1      M21P1 (X1,YMIN)  (X1,Y1)    1.5  SYMM
 JLINE  1    M21P1 MP1   (X1,Y1)    (X1,Y2)    1.3  SYMM
 JLINE  LP1  1      M21P1 (X2,YMIN)  (X2,Y1)    1.5  SYMM
 JLINE  LP1  M21P1 MP1   (X2,Y1)    (X2,Y2)    1.3  SYMM
*
 FILSRF  1   LP1   1      M21P1
 FILSRF  1   LP1   M21P1 MP1
```

```
*
 DOMAIN 3
*
 L  NX3
 M  NY3
*
 ILINE 1    1  LP1  (X2,Y1)    (X3,Y1)  1.0
 ILINE MP1  1  LP1  (X2,Y2)    (X3,Y2)  1.0
 JLINE 1    1  MP1  (X2,Y1)    (X2,Y2)  1.3  SYMM
 JLINE LP1  1  MP1  (X3,Y1)    (X3,Y2)  1.3  SYMM
*
 FILSRF  1   LP1   1   MP1
*
*----
*
 DOMAIN 4
*
 L  NX4
 M  NY4
*
 ILINE 1      1    LP1   (X3,YMIN)    (X4,YMIN)  1.3  SYMM
 ILINE M41P1 1    LP1   (X3,Y1)      (X4,Y1)    1.3  SYMM
 ILINE MP1    1    LP1   (X3,Y2)      (X4,Y2)    1.3  SYMM
*
 JLINE 1      1      M41P1  (X3,YMIN)   (X3,Y1)    1.5  SYMM
 JLINE 1      M41P1 MP1    (X3,Y1)     (X3,Y2)    1.3  SYMM
 JLINE LP1    1      M41P1  (X4,YMIN)   (X4,Y1)    1.5  SYMM
 JLINE LP1    M41P1 MP1    (X4,Y1)     (X4,Y2)    1.3  SYMM
*
 FILSRF  1   LP1   1      M41P1
 FILSRF  1   LP1   M41P1  MP1
*
*----
*
 DOMAIN 5
*
 L  NX5
 M  NY5
*
 ILINE 1   1  LP1  (X4,Y1)    (X5,Y1)   1.3  F
 ILINE MP1 1  LP1  (X4,Y2)    (X5,Y2)   1.3  F
 JLINE 1   1  MP1  (X4,Y1)    (X4,Y2)   1.3  SYMM
 JLINE LP1 1  MP1  (X5,Y1)    (X5,Y2)   1.3  SYMM
*
 FILSRF  1   LP1   1   MP1
*
*----
END
*
*-------------------------------------------------------------------------
*
PROBLEM_TYPE
 SOLVE FLOW_C TURBULENCE HEAT SWIRL
END
*
*-------------------------------------------------------------------------
*
PROPERTIES
 DENSITY GAS_LAW PRESS=0.0  MOL_WT=28.7
 VISCOSITY CONSTANT VISLAM
END
*
*-------------------------------------------------------------------------
*
MODELS
* Uses default turbulence model: k-e with wall functions.
```

```
END
*
*----------------------------------------------------------------------
*
BOUNDARY_CONDITIONS
*
* DOMAIN 1
*
  WALL     1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL       1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  TOTAL_P    1    1    1    M
  U=0.0  V=0.0  W=0.0  P=PUPST T=TUPST K=2.0  D=10000.0
*
  INTERFACE   L   L   1   M
*
*---
*
 DOMAIN 2
*
  WALL     1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
  WALL     1    1    1    M21
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
  WALL     L    L    1    M21
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL       1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  INTERFACE   1    1    M21P1   M
  INTERFACE   L    L    M21P1   M
*
*---
*
 DOMAIN 3
*
  WALL     1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL       1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  INTERFACE   1   1   1   M
  INTERFACE   L   L   1   M
*
*---
*
 DOMAIN 4
*
  WALL   1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
  WALL   1    1    1    M41
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
  WALL   L    L    1    M41
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL       1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  INTERFACE   1    1    M41P1   M
  INTERFACE   L    L    M41P1   M
*
```

```
*---
*
 DOMAIN 5
*
  WALL    1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL      1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  EXIT_P    L    L    1    M
  U=0.0  V=0.0  W=0.0  P=PDOWN T=TDOWN K=2.0  D=10000.0
*
  INTERFACE   1   1   1   M
*
END
*
*-------------------------------------------------------------------
*
INITIAL_CONDITIONS
*  same conditions for all 5 domains.
 U=UINL V=0.0  W=0.0  P=1.1E5  T=300.0  K=2.0  D=1000.0
END
*
*-------------------------------------------------------------------
*
SOLUTION_CONTROL
  ALGORITHM SIMPLEC
  S_SCHEME CENTRAL  U V W
  S_BLENDING 0.1 U V W
  RELAX 0.6 VIS P
  ITERATIONS 400
  SOLVER WHOLE_Y U V W PP K D
  S_ITERATIONS 10 U V W K D
  S_ITERATIONS 25 PP
  C_ITERATIONS 3
  MAXVAL 0.10 VIS
END
*
*-----------------------------------------------------------------------
OUTPUT
 M_FLOW ON
END
```

Sample Input File for Axisymmetric Incompressible Flow in a 2-Cavity Labyrinth Seal (Multi-Domain version)

```
title  ' Incompressible flow labyrinth seal with two cavities '
*-----------------------------------------------------------------------
*
*
* >>  All units below are in SI>  However, any *consistent* set
*     of units can be used in this input file. <<
*
*  PLEASE CHECK THE USERS MANUAL IF ADDITIONAL INFORMATION IS NEEDED.
*
*  Contact Mahesh Athavale at mma@cfd1.cfdrc.com
*                 or call at (205)536-6576
*  if you need something beyond the material here or as a last resort.
*
************************************************************************
*
* Two-Cavity Labyrinth Seal Configuration.
*   incompressible flow
*  The axisymmetric flow is simulated with 2-D grid.
*
*
*     Dom 1.     Dom2      Dom. 3    Dom. 4        Dom. 5
*       ---|----------------|--|-----------------|------
*   P   ---|                |--|                 |------ P
* upstr.   |                |  |                 |        downstr.
*          |                |  |                 |
*          |                |  |                 |
*          |                |  |                 |
*          ------------------   -----------------
*
*  For additional knives, you will need to expand the grid indices,
*    grid generation, blockage and the boundary conditions.
*
*  Current setup is for specified pressures, both upstream and
*   downstream.  Appropriate changes will be needed for
*   other combinations.
************************************************************************
*
*  2-D axisymmetric flow with seal axis along the Cartesian X.
*
*  Multi block grid used for efficiency.
*-----------------------------------------------------------------------
*
PARAMETERS
*
*
*  define teeth thickness and cavity width in terms of cells
*
INTEGER  LTEETH=6,  LCAV=20
INTEGER  MCAV=20,   MGAP=10
*
*  Upstream and downstream of seal:
*
* upstream, domain 1.
 INTEGER NX1=LTEETH, NY1=MGAP
* Cavity No. 1
 INTEGER NX2=LCAV,    NY2=MCAV+MGAP
 INTEGER M21=MCAV,    M21P1=M21+1
*  tooth No. 2
 INTEGER NX3=LTEETH, NY3=MGAP
*  cavity No. 2
 INTEGER NX4=LCAV,    NY4=MCAV+MGAP
 INTEGER M41=MCAV,    M41P1=M41+1
*  downstream
 INTEGER NX5=LTEETH, NY5=MGAP
*
*  dimensions. ---------------------------------------------
*  axial dimensions:  xup, xdown, xcav and xteeth
*  fill up these with dimensiona for your case.
*
 REAL XUP=2.E-3, XDOWN=10.E-3,  XTEETH=2.E-3,  XCAV=10.E-3
*
*  define locations for grid generation
*
 REAL XMIN=0.0,  X1=XMIN+XUP,  X2=X1+XCAV,  X3=X2+XTEETH
 REAL X4=X3+XCAV,  X5=X4+XDOWN
*
*  radial direction, ycav and ygap
*
 REAL YCAV=10.E-3,  YGAP=0.7E-3
*
 REAL YMIN=25.E-3,  Y1=YMIN+YCAV,  Y2=Y1+YGAP
*
*  physical parameters.
*  rpm=rotor speed in rpm.,  densit=density pupst=upstrem pressure
*  pdown=downstream pressure. tupst=upstrem temperature
* pressure in Pa, temp in Kelvin.
*  densit=density kg/m**3, vislam=kinematic viscosity, m**2/s
*
 REAL RPM=4000.,  PI=3.14159, RPS=RPM/60., OMX=2.*PI*RPS
 REAL PUPST=1.4E5,  PDOWN=1.0E5, TUPST=400., TDOWN=TUPST
 REAL DENSIT=1000., VISLAM=1.E-6
 REAL UINL=30.0
*
*
END
*---------------------------------------------------------------------
*
GEOMETRY
 GRID 2D BFC  AXISYMMETRIC
*
* DOMAIN 1
*
 L  NX1
 M  NY1
*
 ILINE 1   1  LP1  (XMIN,Y1)  (X1,Y1)    1.0
 ILINE MP1 1  LP1  (XMIN,Y2)  (X1,Y2)    1.0
 JLINE 1   1  MP1  (XMIN,Y1)  (XMIN,Y2)  1.3  SYMM
 JLINE LP1 1  MP1  (X1,Y1)    (X1,Y2)    1.3  SYMM
*
 FILSRF  1   LP1   1   MP1
*
*---------
*
 DOMAIN 2
*
 L  NX2
 M  NY2
*
 ILINE  1     1  LP1  (X1,YMIN) (X2,YMIN)  1.3  SYMM
 ILINE  M21P1 1  LP1  (X1,Y1)   (X2,Y1)    1.3  SYMM
 ILINE  MP1   1  LP1  (X1,Y2)   (X2,Y2)    1.3  SYMM
*
 JLINE  1    1     M21P1 (X1,YMIN)  (X1,Y1)    1.5  SYMM
 JLINE  1    M21P1 MP1   (X1,Y1)    (X1,Y2)    1.3  SYMM
 JLINE  LP1  1     M21P1 (X2,YMIN)  (X2,Y1)    1.5  SYMM
 JLINE  LP1  M21P1 MP1   (X2,Y1)    (X2,Y2)    1.3  SYMM
*
 FILSRF  1   LP1   1     M21P1
 FILSRF  1   LP1   M21P1 MP1
*
*----
```

```
*
 DOMAIN 3
*
 L  NX3
 M  NY3
*
 ILINE 1    1  LP1  (X2,Y1)   (X3,Y1)  1.0
 ILINE MP1  1  LP1  (X2,Y2)   (X3,Y2)  1.0
 JLINE 1    1  MP1  (X2,Y1)   (X2,Y2)  1.3  SYMM
 JLINE LP1  1  MP1  (X3,Y1)   (X3,Y2)  1.3  SYMM
*
 FILSRF  1   LP1   1   MP1
*
*----
*
 DOMAIN 4
*
 L  NX4
 M  NY4
*
 ILINE 1      1     LP1   (X3,YMIN)   (X4,YMIN)  1.3  SYMM
 ILINE M41P1  1     LP1   (X3,Y1)     (X4,Y1)    1.3  SYMM
 ILINE MP1    1     LP1   (X3,Y2)     (X4,Y2)    1.3  SYMM
*
 JLINE 1      1      M41P1  (X3,YMIN)   (X3,Y1)    1.5  SYMM
 JLINE 1      M41P1  MP1    (X3,Y1)     (X3,Y2)    1.3  SYMM
 JLINE LP1    1      M41P1  (X4,YMIN)   (X4,Y1)    1.5  SYMM
 JLINE LP1    M41P1  MP1    (X4,Y1)     (X4,Y2)    1.3  SYMM
*
 FILSRF  1   LP1   1      M41P1
 FILSRF  1   LP1   M41P1  MP1
*
*----
*
 DOMAIN 5
*
 L  NX5
 M  NY5
*
 ILINE 1   1  LP1  (X4,Y1)   (X5,Y1)   1.3  F
 ILINE MP1 1  LP1  (X4,Y2)   (X5,Y2)   1.3  F
 JLINE 1   1  MP1  (X4,Y1)   (X4,Y2)   1.3  SYMM
 JLINE LP1 1  MP1  (X5,Y1)   (X5,Y2)   1.3  SYMM
*
 FILSRF  1   LP1   1   MP1
*
*----
END
*
*-----------------------------------------------------------------------
*
PROBLEM_TYPE
 SOLVE FLOW TURBULENCE SWIRL
END
*
*-----------------------------------------------------------------------
*
PROPERTIES
 DENSITY CONSTANT DENSIT
 VISCOSITY CONSTANT VISLAM
END
*
*-----------------------------------------------------------------------
*
MODELS
* Uses default turbulence model: k-e with wall functions.
```

```
END
*
*------------------------------------------------------------------------
*
BOUNDARY_CONDITIONS
*
* DOMAIN 1
*
  WALL    1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL      1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  EXIT_P    1    1    1    M
  U=0.0  V=0.0  W=0.0  P=PUPST T=TUPST K=2.0  D=10000.0
*
  INTERFACE   L   L   1   M
*
*---
*
 DOMAIN 2
*
  WALL    1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
  WALL    1    1    1    M21
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
  WALL    L    L    1    M21
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL      1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  INTERFACE   1    1    M21P1    M
  INTERFACE   L    L    M21P1    M
*
*---
*
 DOMAIN 3
*
  WALL    1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL      1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  INTERFACE   1   1   1   M
  INTERFACE   L   L   1   M
*
*---
*
 DOMAIN 4
*
  WALL    1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
  WALL    1    1    1    M41
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
  WALL    L    L    1    M41
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL      1    L    M    M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  INTERFACE   1    1    M41P1    M
  INTERFACE   L    L    M41P1    M
*
```

```
*---
*
 DOMAIN 5
*
  WALL    1    L    1    1
  U=0.0  V=0.0  WA=OMX  K=0.0  D=0.0
*
  WALL      1     L     M     M
  U=0.0  V=0.0  W=0.0  K=0.0  D=0.0
*
  EXIT_P    L     L     1     M
  U=0.0  V=0.0  W=0.0  P=PDOWN T=TDOWN K=2.0  D=10000.0
*
  INTERFACE   1   1   1   M
*
END
*
*------------------------------------------------------------------
*
INITIAL_CONDITIONS
*  same conditions for all 5 domains.
 U=UINL V=0.0  W=0.0  P=1.1E5  T=300.0  K=2.0  D=1000.0
END
*
*------------------------------------------------------------------
*
SOLUTION_CONTROL
  ALGORITHM SIMPLEC
  S_SCHEME CENTRAL  U V W
  S_BLENDING 0.1 U V W
  RELAX 0.6 VIS P
  ITERATIONS 400
  SOLVER WHOLE_Y U V W PP K D
  S_ITERATIONS 10 U V W K D
  S_ITERATIONS 25 PP
  MAXVAL 0.10 VIS
END
*
*-----------------------------------------------------------------------
OUTPUT
 M_FLOW ON
END
```

APPENDIX F

EXECUTION OF THE CODES

APPENDIX F

EXECUTION OF THE CODES

After an input file has been prepared, either through the use of a text editor or the GUI, several additional steps are needed to successfully execute the code. Following is a brief description of these steps. The description is for a UNIX based operating system, since a vast majority of the workstations and mainframe computers use UNIX. However, sufficient information is also included to enable the user to run the code on other operating systems. In the following, **SCIPRE** and **SCISEAL** correspond to the executables of the pre-processor and the flow solver, respectively.

F.1 Execution of Pre-Processor

After the text input file is created, the next step is to run the pre-processor **SCIPRE** to convert this information in a form suitable for **SCISEAL**. To do this the **SCIPRE** code should be executed with the text input file as the default input unit (unit = 5 in FORTRAN). On a UNIX based system this can be accomplished simply by using the command:

```
$ SCIPRE < problem.in
```

where **problem.in** is the name of the text input file. The pre-processor has several built in checks for data consistency and will print out error messages if any exist in the input file. These messages are written to the default output unit (unit = 6). If not specified otherwise, these messages are printed to the terminal. These can also be 'piped' to an output file on a UNIX system by modifying the pre-processor execution command:

```
$ SCIPRE < problem.in > pre.out
```

when the pre-processor is successfully executed, it creates several files in the default directory. Two files that are essential for **SCISEAL** execution are:

a. **model.DAT**: text file that contains all information for the problem including boundary conditions, problem type, discretization schemes, etc.

b. **model.AUG**: this contains the information on the grid size as well as the grid coordinates. By default it is a binary file with double precision real numbers. Binary files require less space on the disk and double precision accuracy is often needed for the small clearances seen in seal applications.

Several other files that may also need to be printed out are **model.INT, model.VAR,** and **model.WAL** depending on the type of the problem and output information requested. A description of these files is given in Appendix C.

If the operating system being used is other than UNIX, then the I/O units will probably need to be defined explicitly before **SCIPRE** execution. The steps should involve:

a. definition of I/O unit 5 as file input with **problem.in** as the disk file
b. definition of **pre.out** as the default unit = 6 (optional)

Then **SCIPRE** can be run the appropriate system command for execution. Most operating systems will allow such external definition of the I/O units for a FORTRAN code, but the specific form of the command may vary.

F.2 Execution of the Flow Solver

The execution of the main code requires at least two files in the default directory: **model.DAT** and **model.AUG** which are created by the pre-processor. In addition, **model.AUR**, the restart data file may be needed if the RESTART option used in the input file for the continuation of a previously completed run. Most of the text output from **SCISEAL** is directed to the default FORTRAN output unit (unit = 6), and it is recommended that this be directed to a disk file to keep a record of the execution. For a UNIX based system, this is simply done by using the command:

```
$ SCISEAL > & model.out
```

The & in this command directs any system error messages to the output file as well. At the completion of the run the code will create a binary restart file **model.AUR** which can be used for a continuation run if needed. Convergence history for the run is output in a file named **model.RSL**. If data output in PLOT3D form is requested, these files are also created at every intermediate data dump as well as at the end of the run.

INDEX

COMMAND	DESCRIPTION	PAGE
LPDREF	domain number containing the pressure anchor point	9-15
LPPREF	index of cell for anchoring pressure	9-15
M	maximum number of cells in the domain in the j grid index	5-6
MAXVAL	maximum value trap for a variable	11-8
MINVAL	minimum value trap for a variable	11-8
N	maximum number of cells in a domain in the k grid index	5-6
PLOT3D	command to activate writing of PLOT3D files	12-5
POINT	command to identify points using specified names and coordinates	5-11
POLYNOMIAL	command to specify a polynomial curve for properties	5-10
PRINT	command to activate parameter values	4-2
PRINTF	command to activate field print of variables at the end of a run	8-2
PROJECT_I	generation of 3-D grid by projecting 2-D surface in i direction	5-17
PROJECT_J	generation of 3-D grid by projecting 2-D surface in j direction	5-17
PROJECT_K	generation of 3-D grid by projecting 2-D surface in k direction	5-17
PRL	turbulent Prandtl number	7-6
READ GRID	command to access grids from external file	5-4
READ GRID FROM	command to read grid from an external file	5-19
REAL	parameter type declaration for real variables	3-5
RELAX	direct relaxation for variables during iterations	11-6
RESPRN	command to control frequency of residual printing	12-3
RESTART FROM	command to read initial conditions from previous run	10-2
RESTART_NAME	name of the file to save information for restart run	12-2
RESTART_SAVE	set the frequency at which restart files are saved	12-3
REVOLVE_I	command to generate volume grid by revolution of 2-D grid about a specified axis in the i direction	5-18
REVOLVE_J	command to generate volume grid by revolution of 2-D grid about a specified axis in the j direction	5-18
REVOLVE_K	command to generate volume grid by revolution of 2-D grid about a specified axis in the k direction	5-18
RHOREF	reference fluid density for natural convection	7-3
RLOAD	command for rotor pressure load	12-6
ROTATE	command to activate a transfer to a rotating reference frame	5-19

COMMAND	DESCRIPTION	PAGE
WALL_R	rotating wall boundary specification in 3-D	9-9
XGRID	grid generation in x direction for cartesian grids	5-6
YGRID	grid generation in y direction for cartesian grids	5-6
ZGRID	grid generation in z direction for cartesian grids	5-6

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 2005	Final Contractor Report

4. TITLE AND SUBTITLE

SCISEAL: A Computer Program for Study of Fluid Dynamic Forces in Seals
Users' Manual

5. FUNDING NUMBERS

WU–506–42–31–00
WU–590–21–11–00
NAS3–25644
MTI Subcontract 004–05022

6. AUTHOR(S)

M.M. Athavale and A.J. Przekwas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

CFD Research Corporation
215 Wynn Drive
Huntsville, Alabama 35805

8. PERFORMING ORGANIZATION REPORT NUMBER

E–13619

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR—2003-212357
CFDRC PR 4116/1

11. SUPPLEMENTARY NOTES

M.M. Athavale and A.J. Przekwas, CFD Research Corporation, 3325 Triana Boulevard, Huntsville, Alabama 35805. Project Manager, Anita D. Liang, Aeropropulsion Projects Office, NASA Glenn Research Center, organization code P, 216–977–7439.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Restriction changed to Unclassified/Unlimited on July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

~~**Export Administration Regulations (EAR) Notice**~~

~~This document contains information within the purview of the Export Administration Regulations (EAR), 15 CFR 730–774, and is export controlled. It may not be transferred to foreign nationals in the U.S. or abroad without specific approval of a knowledgeable NASA export control official, and/or unless an export license/license exception is obtained/available from the Bureau of Industry and Security, United States Department of Commerce. Violations of these regulations are punishable by fine, imprisonment, or both.~~

Unclassified-Unlimited
Subject Category: 34
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Seals Scientific Code (SCISEAL) is an advanced Computational Fluid Dynamics (CFD) package, designed for the calculations of turbulent flows and fluid dynamic forces in different types of seals such as cylindrical, labyrinth, face, and tip seals. SCISEAL is an implicit multi-domain code and its basic algorithms can handle colocated variables, moving grids, rotating coordinates, etc. This report provides guidance to the users of SCISEAL in problem definition and input file preparation and describes the code structure, model and grid generation, boundary conditions, and several seal specific features. Sample input files are also provided. The SCISEAL code is advanced and general so that it can be used to compute flow in a wide variety of flow problems besides seals. This manual provides a fairly detailed description of the general parameters needed to set up input files and provides some guidelines for the user to avoid commonly made mistakes and provides some tips for obtaining a solution and optimizing code performance.

14. SUBJECT TERMS

Seals; Face seals; Cylindrical seals; Incompressible; Gas lubricated; Leakage; Users' manual; Fluid dynamics; CFD; Computer code; Analysis; Compressible

15. NUMBER OF PAGES

170

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	